AÑO II - NUM. 26

MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO II- N.º 26

95 PTAS.

Canarias 105 ptas.

HOP EDITA HOBBY PRESS, S.A.

PROGRAMAS

INTRUDER

BUSCA PAR

CAMIONERO

CALCULADORA

MICROPANORAMA

ESPECIAL INFORMAT 85

BASIC

DEPURACION DE PROGRAMAS

ESTRATEGIA

ESTRUCTURA Y CREACION DE LABERINTOS

"HOBBY SUERTE"

¡250.000 pts. en premios cada semana!

ESTE NUMERO PUEDE SIGNIFICAR UN FABULOSO REGALO PARA TI.

74655356

¡Consulta a tu Spectrum!

Cada semana, Microhobby regala 70 premios entre sus lectores. La clave del premio es el número que figura en este cupón, en la esquina superior derecha.

Para saber si el número de tu ejemplar está premiado, debes introducirlo en tu Spectrum, utilizando para ello el programa «Hobby-Suerte». La cassette con este programa se ha entregado a los lectores junto con el número 15 de Microhobby Semanal. Si no posees esta cinta, puedes pedir una copia a un amigo o por carta a **Hobby Press, S.A., Apartado n.° 54062. Madrid,** incluyendo dentro del sobre 180 pts. en 3 sellos de Correos de 60 pts. cada uno. Este programa sirve para leer todos los números durante las **50 semanas** que dure este Concurso.

(Bases en el reverso)

Premios semanales

■ Primera Categoría

Un **Spectrum 48 k.** (o un Microdrive y un Interface 1, a elegir por el interesado).

■ Segunda Categoría

Una **Impresora GP 50 de Seikosha,** especialmente diseñada para Spectrum (2 premios).

■ Tercera Categoría

Un **Joystick con su interface** (3 premios).

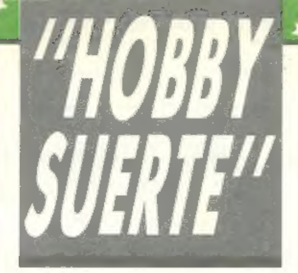
■ Cuarta Categoría

Una **Suscripción a Microhobby Semanal** por un año (50 números. Si el lector premiado ya es suscriptor, podrá optar por prolongar su suscripción anual o un premio de Quinta Categoría) (14 premios).

■ Quinta Categoría

Una **Cinta de Programa,** a elegir entre un variado surtido de juegos, utilidades, etcétera (50 premios).

Hobby Press, S.A. garantiza que cada semana introduce al azar, entre todos los ejemplares que componen la edición, **setenta cupones** correspondientes a los premios aquí citados.



Instrucciones para concursar

Una vez introducido en memoria el programa **«Hobby-Suerte»,** aparecerá en la pantalla la clásica máquina «tragaperras» de frutas.
A continuación, debes teclear el número que figura en esta tarjeta. Al pulsar «Enter», la «máquina» se pone en marcha y te hace saber si has sido agraciado con uno de los 70 premios semanales.

Muy importante: Puede ocurrir que, al introducir en el programa números al azar, alguno de ellos corresponda casualmente a un premio. También es posible, con los suficientes conocimientos de Basic, alterar el programa para que un número determinado aparezca como premiado.

Por todo ello, debemos aclarar los siguientes aspectos:

1. El único justificante para reclamar un premio determinado es la posesión del cupón con el número impreso en él.

2. Todos los números susceptibles de dar premio están registrados ante Notario.

3. Hobby Press, S.A. no se hace responsable de ningún otro cupón que no corresponda a los números previamente registrados. Tampoco se atenderán reclamaciones verbales que no vengan acompañadas por la posesión del cupón con el número premiado.

4. Cualquier lector puede solicitar de esta Editorial la comprobación de la entrega de los Premios semanales.

5. Hobby Press, S.A. se reserva el derecho a resolver según su criterio cualquier cuestión no prevista en las Bases de este Concurso.

6. La reclamación de cualquier Premio de este Concurso caduca el día 30 de Junio de 1986.

COMUNICACION DE PREMIO
(Enviar rellenado con letra clara y en sobre cerrado)

Nombre .... Edad ....
Apellidos ....
Domicilio .... Teléfono ....
Ciudad .... C.P. .... Provincia ....
Categoría del Premio Obtenido .... Número de Microhobby ....

Si consideras que tu cupón tiene premio, fotocópialo como medida de seguridad y envíalo por **Correo Certificado** a Hobby Press, S.A., Apartado 54.062 de Madrid. Por favor, anticípanos todos estos datos por teléfono, llamando al (91) 654 32 11. En este mismo número atenderemos cualquier consulta o duda sobre las Bases o la mecánica de «Hobby Suerte».

Envía este cupón por correo certificado a HOBBY PRESS. Apt. 54.062. Madrid

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Subdirector
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Jesús Iniesta
Maqueta
Rosa María Capitel
Redacción
José María Díaz, Miguel Angel Hijosa, Fco. Javier Martín
Colaboradores
Jesús Alonso, Lorenzo Cebeira, Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce
Dibujos
Manuel Berrocal, J.R. Ballesteros, A. Perera, F.L. Frontán, J. Septien, Pejo, J.M. López Moreno

Edita
HOBBY PRESS, S.A.

Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Administrador General
Ernesto Marco

Jefe de Publicidad
Marisa Esteban
Secretaria de Publicidad
Concha Gutiérrez
Publicidad Barcelona
Isidro Iglesias
Tel.: (93) 307 11 13

Secretaria de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, n.º 8
Polígono Industrial de Alcobendas
Tel.: 654 32 11

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S.A. Valencia, 245
Barcelona

Imprime
Rotedic, S.A.
Carretera de Irún, Km. 12,450
Tel.: 734 15 00
Fotocomposición
Espacio y Punto, S.A.
Paseo de la Castellana, 268
Fotomecánica
Lasercolor
Alejandro Villegas, 31
Depósito Legal:
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América, 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO II. N.º 26. 30 de abril al 6 de mayo de 1985
95 ptas. (Sobretasa Canarias 10 ptas.)

PREMIADOS HOBBY-SUERTE

ESTA SEMANA

JUAN A. GARCIA OLIVER, Alos, 24 (PALMA DE MALLORCA).
Suscripción a Microhobby Semanal por una año (4.º Cat.)

JESUS MANGUILLO BIENDICHO, Avda. Casaldo, 84, 5.º (CASTELLON DE LA PLANA).
Cinta de programas (5.º Cat.)

CARLOS SANCHEZ MINGUEZ, Trabajo, 2. Puerta de Sagunto (VALENCIA).
Cinta de programas (5.º Cat.)

MANUEL IGLESIAS LOPEZ, Avda. Manzanares, 62, 2.º B (MADRID).
Cinta de programas (5.º Cat.)

JAVIER GARCIA ESCOBAR, Teruel, 9, 2.º I (MADRID).
Cinta de programas (5.º Cat.)

JESUS RUIZ VIGO, Maestro Portela, 36, 2.º D. San Fernando (CADIZ).
Cinta de programas (5.º Cat.)

BARBARA FRANCISCA SANCHEZ, Marroquina, 94, 2.º D (MADRID).
Un Joystick con su interface (3.º Cat.)

JUAN IGUAL ALONSO, Poblado de C.N.V., 35, Valdecaballeros (BADAJOZ).
Suscripción a Microhobby Semanal por un año (4.º Cat.)

MIGUEL PASTOR SEBA, Pilar Martí, 16, 9.º, Burjasot (VALENCIA).
Cinta de programas (5.º Cat.)

ANDRES COLMENERO TORNERO, D'anton y Conca, 4, 1.º, Onteniente (VALENCIA).
Un Joystick con su interface (3.º Cat.)

RAFAEL LOZANO GARCIA, Luis de Hoyo Saínz, 180, 5.º C (MADRID).
Cinta de programas (5.º Cat.)

JOSE M.ª LLUESMA SANCHEZ, Camino de Vinateros, 4, 2.º A, Moratalaz (MADRID).
Suscripción a Microhobby Semanal por un año (4.º Cat.)

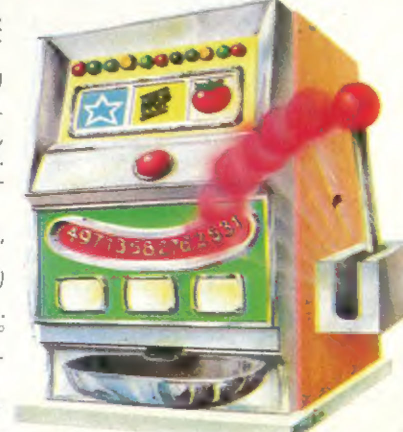
FRANCISCO J. GOMEZ RUIZ, Coral, 6, 3.º I (SEVILLA).
Cinta de programas (5.º Cat.)

JOSE LUIS BUSTAMONTE S. MIGUEL, Constitución, 49, 1.º A, Sama de Langrea (ASTURIAS).
Cinta de programas (5.º Cat.

JOSE IGNACIO RUIZ OCARIZ, Paseo de Zorrilla, 144, 3.º I (VALLADOLID).
Cinta de programas (5.º Cat.)

MANUEL TAPIA VARGOLA, C. López, 6, Santa Amalia (BADAJOZ).
Suscripción a Microhobby Semanal por un año (4.º Cat.)

LUIS SILLA VALLEJO, Galileo, 76, 4.º B, Arguelles (MADRID).
Cinta de programas (5.º Cat.)



MICROPANORAMA

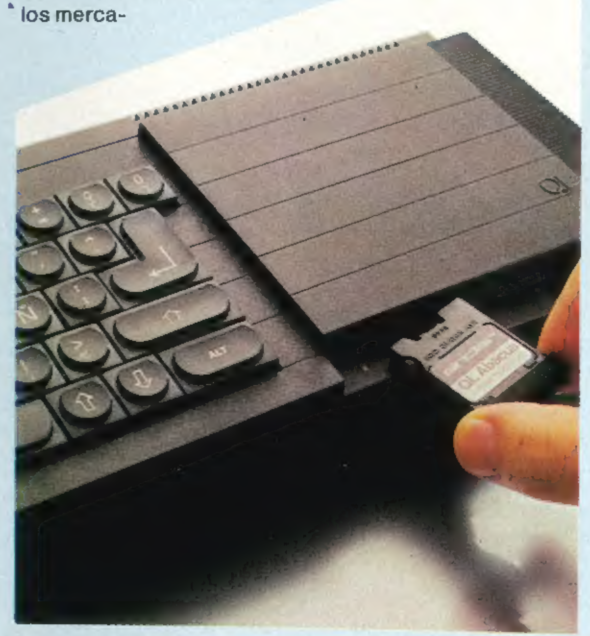
ESPECIAL INFORMAT 85

EL «QL» A LA ESPAÑOLA

Por fin, y después de mucho tiempo de espera, ha llegado el QL versión española, o al menos ha sido presentado en el Informat, la feria de Informática de Barcelona, aunque los responsables del tema nos confirmaron que tendrán que pasar todavía unas tres semanas para que se encuentre disponible en los comercios.

Se trata de la primera vez que Sinclair Research LTD, realiza una versión local, con la colaboración de su distribuidor en España, Investrónica, adelantándose de este modo a otros países que, en un futuro, presentarán también sus versiones locales.

En total aparecerán 12 versiones en diferentes lenguas de las cuales la española ha sido la primera de todas. Esta política se sigue en un intento claro de expandir los ordenadores personales en los merca-

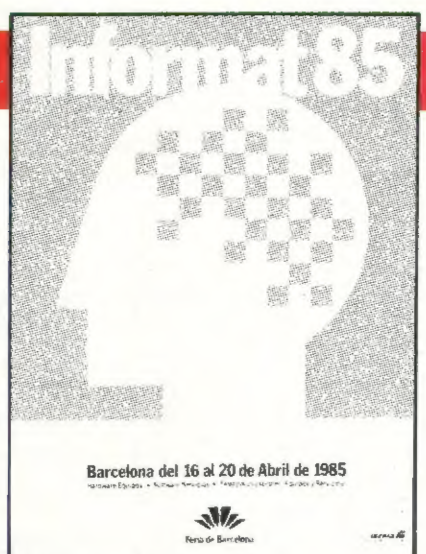


dos profesionales y de negocios en todo el mundo para intentar, de esta forma, reforzar la posición de la compañía de liderazgo mundial en este mercado.

La versión española del QL incorpora interesantes avances y, además, algunas actualizaciones dentro del sistema como por ejemplo, el hecho de que hayan sido traducidos todos los mensajes del sistema, los avisos de error y la incorporación de los caracteres castellanos, con los que no contaban los ordenadores de procedencia anglosajona, como son los siguientes: «¡ ¿ Ñ ñ u C».

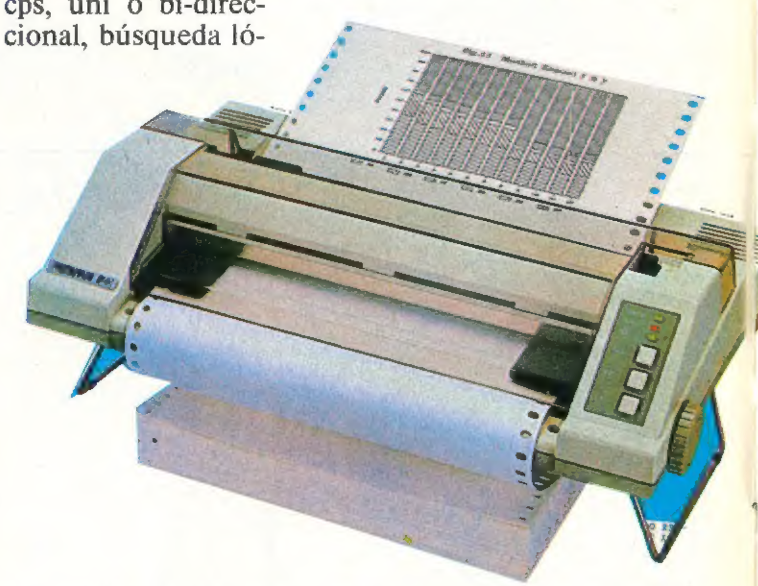
Los programas que acompañan al QL, han sido también traducidos.

El precio de este aparato será de 125.000 ptas. y sus compradores pasarán automáticamente, previo pago, eso sí, de una cuota mensual, a formar parte del QLUB, un ente que a través de un boletín informativo y una línea telefónica, les proporcionará actualizaciones de Software, noticias, notas técnicas y además, atenderá todo tipo de consultas relacionadas con este ordenador.



LA «SUPERIMPRESORA» RITEMAN FT

La impresora Riteman F+, reúne una serie de condiciones que la hacen atractiva de cara al usuario de ordenador. Tiene una volocidad de impresión de 105 cps, uni o bi-direccional, búsqueda ló-



gica optimatizada, capacidad de impresión de 45 Ipm a 10 cpi, y 200 ms de salto de línea y espacio de 1/6 pulgadas.
standard, el de doble impresión, enfatizado, itálica, supra y subíndices, mitad altura y el NLQ.

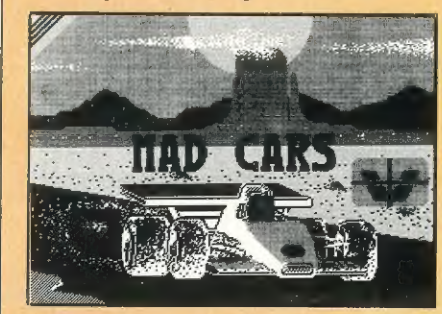
Otra de las características de esta impresora son: el espacio interlíneas, corte de papel, selección de caracteres, test de escritura, subrayado contínuo, tabulador, buffer de 2 Kbytes (standard) y 8 Kbytes (opcional), tope posicionador de papel contínuo y cinta impresora autorretintada.

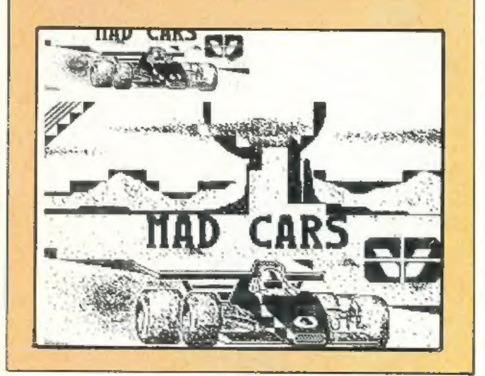
Utiliza un Interface standard paralelo centronics de 8 bits y uno en serie RS232 C, opcional.

UNA RUTINA PARA MEJORAR LOS COPYS

Pin Soft ha desarrollado una serie de rutinas de adecuación para las impresoras Riteman F+ y Seikosa SP-800, que permite hacer Copys de pantalla convirtiendo los colores originales en diferentes tonalidades de grises acordes con su luminosidad, es decir, al igual que se veía un televisor en B/N.

En una impresora normal, sin esta rutina, la tinta siempre es interpretada como negro y el papel como blanco, con el consiguiente deterioro de la imagen impresa con respecto a la original.





WRIGGLER: EL GUSANO

La empresa española de software BABETA, estuvo presente en el Informant con un programa nuevo que ha comenzado a distribuir en España. Es de la casa Romantic Robot y tiene como protagonista a un simpatico personaje: una especie de gusano que tiene que recorrer peligrosos caminos en un maratón poco habitual. El personaje en cuestión, se vende con el juego.



BOXER 12, EL NUEVO MONITOR DE HANTAREX

Hantarex, una casa especializada en monitores, ha presentado el Boxer 12, un monitor monocromo de 12 pulgadas, con una señal de vídeo compuesto, que podría conectarse al Spectrum mediante un Interface espacial.

Tiene una amplitud de banda de 20 MHz y una resistencia de entrada de 75 Ohm. Existe la posibilidad de audio y sus medidas le hacen muy cómodo de instalar en cualquier lugar, 309 por 280 mm y una anchura de 264 mm.

Puede llevar una pantalla antirreflectante y tiene un panel, en la parte frontal, extraíble, donde se encuentran los botones para ajustar el aparato.



IDEALOGIC: NUEVO RUMBO

Idealogic ha emprendido una nueva línea con respecto a los programas que normalmente viene lanzando al mercado. Hasta ahora, los programas que venían comercializando, estaban dirigidos a niños hasta los 13 años e incluso menos. Los nuevos programas son para jóvenes desde 13 años, en adelante. Son una serie de juegos de Misterio, Ciencia Ficción y Novelas Clásicas donde el usuario es el protagonista.

Para poder jugar con ellos es necesario leer un pequeño libro en el que se explican las características del personaje. Con ello se pretende, además de instruir al jugador, que éste conozca el vocabulario, condición indispensable para jugar a los juegos de aventuras. Los programas son en castellano y, de momento, sólo han sido disponibles para Commodore, aunque muy pronto lo estarán también para Spectrum.

Son «Profesión Detective», «La Familia Robinson», y «Una Fantástica Aventura».

En el stand de Idealogic encontramos además un monitor para QL, que además es compatible para otros ordenadores, y una Tortuga para Spectrum.

ACUERDO ABC - ULTIMATE



La empresa ABC Analog ha firmado una serie de acuerdos con varias compañías de conocido renombre, tanto nacional como internacional. El más destacado de todos ha sido el realizado con la archiconocida (ULTIMATE), la empresa de software más popular en los últimos tiempos en el Reino Unido. Por este acuerdo, ABC fabricará en España todos sus programas. Para abril están previstos Sabre Wulf, Knight Lore, Underwulde y Alien 8. Los programas Ultimate para Spectrum serán comercializados conjuntamente con otra compañía, Investrónica, y su precio al fabricarse en nuestro país se verá reducido considerablemente (costará unas 2.100 ptas.).

En el Informat pudimos ver el master del Alien 8 recién llegado de Inglaterra.

En el apartado de Hardware ABC presentaba a un viejo conocido, el Currah Speech, el Lápiz óptico de DK'Tronics, un Interface Moden y el Joystick Powerplay.

TRUCOS



TINTA INVISIBLE

Como en las películas de espionaje, podremos escribir mensajes con tinta invisible y así evitar que sean leídos por el «enemigo», con tan sólo poner en práctica un truco que nos envía David García López. Para ello hemos de teclear la sentencia POKE 23607,100 seguido de ENTER.

Al principio te saldrá la pantalla en blanco, pero pulsando SPACE, aparecerá un cuadrado negro, en este momento podrás teclear tus programas sin temor de que nadie los vea.

Si quieres volver a la posición normal, teclea POKE 23607,60 y verás de nuevo lo que escribes.

PARA DESTACAR EN PANTALLA

Javier Guixá Catalá nos manda un truco para conseguir un curioso efecto tridimensional, aprovechando la poca definición de un televisor color doméstico.

Consiste en escribir las líneas sobre los puntos de la pantalla que no quedan totalmente cubiertos por un segmento de color.

Por ejemplo, para obtener una pirámide, introducir las siguientes instrucciones:

```
  10 BORDER 0: PAPER 0: CLS
  20 FOR i=0 TO 87 STEP 2
  30 LET j=i-7*INT (i/7)+1
  40 INK j: PLOT 40+i,i
  50 DRAW 175-2*i,0: DRAW 0,175-
2*i
  60 DRAW 2*i-175,0: DRAW 0,2*i-
175
  70 NEXT i
```

CONSERVACION E IDENTIFICACION DE PROGRAMAS

Una forma de trabajo útil para conservar programas, es tener la buena costumbre de poner, en la primera línea, un «REM» seguido de título entrecomillado, lo que también es aprovechable para obtener su rápida identificación tanto en pantalla como en listado por impresora.

Si editamos la primera línea con el nombre y borramos el número de línea más «REM» poniendo en su lugar SAVE, nos hemos ahorrado teclear el nombre y, además, evitamos los posibles errores que pudiéramos cometer al escribirlo.

Este útil truco, lo envía E. J. Serrano Expósito.

SALTO EN UNA LINEA Y EN UNA SENTENCIA

En el manual del Spectrum, se hace referencia a la posibilidad de hacer un salto entre sentencias. Para R. Domingo Gómez, que nos ha mandado este truco, esto es posible a dos variables del sistema: NEWPPC, línea a la que hay saltar, y NSPPC, sentencia a la que salta.

El salto, consiste en ejercer un POKE a las dos variables con el número de línea y el de sentencia dentro de la línea, de la forma:

POKE 23618, (NUM. de LINEA)-256*INT(NUM. de LINEA/256).

POKE 23619, INT (NUM. de LINEA/256)

POKE 2362Ø, NUM. de SENTENCIA.

En modo programa, podemos usarlo simplemente realizando un GOTO a la línea de programa donde lo hayamos colocado. Como es sabido, conviene colocarlos al principio del listado para una ejecución más rápida.

PROTEGE TU PROGRAMA

Con POKE 23755,X (si X es 1ØØ) podremos hacer desaparecer el programa en BASIC, no pudiendo ser listado, según un truco que nos manda J. Mateos Lago. Si X es Ø, el programa aparecerá y puede ser listado.

MULTIPLES DIBUJOS

Autor del truco anterior, Joaquín Mateos, nos manda otro más con el que hacer dibujos interesantes mediante la rutina:

PLOT 128,5Ø: DRAW OVER 1;Ø,1ØØ,X*PI, donde X es un número impar. Joaquín nos aconseja que esta rutina esté entre 2Ø51 y 2535, aunque se puede probar con otros valores.

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

CALCULADORA

Javier ALEMAN

Spectrum 48 K

Con este programa podremos utilizar nuestro ordenador como una auténtica calculadora con la que realizar cualquier tipo de funciones y recopilación de datos.



El manejo és el siguiente: La calculadora funciona según el sistema conocido como «notación polaca inversa». En este sistema los cálculos son expresados con los operandos en primer lugar, seguidos de la operación. Así, para poner «3+2», en notación polaca inversa se pondría «3 2 +». En nuestra calculadora, a su vez, se calcularía poniendo "3","enter","2","+".

Cuando pulsamos la tecla de una función binaria (por ejemplo la suma) se efectúa dicha operación entre el primer elemento de la pila y el display, quedando el resultado en el display, y desplazándose todos los elementos de la pila un lugar hacia arriba.

Un ejemplo puede aclarar mucho. Supongamos que queremos calcular 1n((2+2)*(5+8)). En nuestra calculadora pondríamos: 2,enter,2,+,5,enter,8,+,*1n.

Pasemos a ver entonces algunas particularidades del uso de nuestra calculadora:

ENTRADA DE DATOS: Para entrar un dato en el display basta digitarlo, como en una calculadora usual. Si nos equivocamos podemos pulsar la tecla «X» (CLEAR) y volver a introducirlo. El punto decimal se encuentra en la «M», y se puede utilizar también notación exponencial con la tecla «E». Para poner el signo «—» para los números negativos, se utiliza la tecla «A».

FUNCIONES: Pueden clasificarse en: *Binarias:* Realizan la función entre el primer elemento de la pila y el display, en ese orden. Son las habituales, +, —, ^, *, /. *Monarias:* Se activan, bien apretando una sola tecla (Q=SIN, W=COS, E=TAN, R=INT, Z=LN), o bien apretando antes la tecla «P» de «función segunda» (Q=ASN, W=ACS, E=ATN, H=SQR, X=EXP, M=PI). Todas ellas actúan sobre el display y tienen el significado habitual sobre una calculadora. *Especiales:* Además están una serie de funciones especiales: CLEAR(X): Borra el contenido del display. LOG(L): Realiza el logaritmo decimal del display. AYUDA(Y): Pantalla de ayuda. NEXT(N): Mueve circularmente la pila. OFF(O): Inicializa la calculadora. STORE(S): Guarda el display en la memoria cuyo número se indique a continuación. DATA(D): Extrae de la memoria indicada su contenido y lo transfiere al display. F. SEGUNDA(P): Para tener acceso a las funciones segundas.

Aparte de las anteriores instrucciones, en caso de haber algún problema se recupera la calculadora con los datos sin cambiar, ejecutando GOTO 200.

En el programa es importante hacer notar que cada vez que aparecen palabras tales como «SIN», «LN», etc., tienen que ser introducidas con una sola tecla, no letra a letra.

```
   1 REM - XAVIER ALAMAN -
  10 POKE 23658,8: DIM c(26): DI
M l$(32): DIM m(10): DIM p(6)
  15 DATA 127,32,88,46,65,76,65,
77,65,78,2600,2500,500,2900,3100
,500,500,2500,500,2500,2500,3100
,500,4600,3020,4100,3100,3100,28
00,500,500,2500,3100,3000,5000,3
100
  16 LET d$="": FOR i=1 TO 10: R
EAD v: LET d$=d$+CHR$ v: NEXT i
  17 FOR i=1 TO 26: READ c(i): N
EXT i
  20 FOR i=1 TO 10: LET m(i)=0:
NEXT i
  25 LET n$="QWERAHJKVBMZQWEHXM"
: LET w$="SIN COS TAN INT -↑-+/*
.LN ASN ACS ATN SQR EXP PI"
  30 FOR i=1 TO 6: LET p(i)=0: N
EXT i
  40 LET ord=0: LET s=0: LET cor
rpila=0
 100 GO SUB 1000: GO SUB 2700: G
O SUB 1710
 200 GO SUB 1500
 210 IF ord=0 THEN LET b$=a$: GO
 SUB 1760: PRINT AT 3,3;b$: LET
s=0: GO SUB 2000
 300 GO SUB a: GO TO 200
 500 RETURN
 990 REM ■ pantalla ■
1000 BORDER 0: PAPER 6: CLS
1010 PAPER 1: PRINT l$;AT 21,0;l
$
1020 FOR i=0 TO 21: FOR j=0 TO 3
1 STEP 31
1030 PRINT AT i,j;" "
1040 NEXT j: NEXT i
1050 INK 0: PLOT 15,135: DRAW 0,
25: DRAW 105,0: DRAW 0,-25: DRAW
 -105,0
1060 FOR i=23 TO 143 STEP 24
1070 PLOT 23,i: DRAW 0,9: DRAW 8
9,0: DRAW 0,-9: DRAW -89,0
1080 NEXT i
1090 PLOT 127,33: DRAW 0,127: DR
AW 113,0: DRAW 0,-127: DRAW -113
,0
1100 PAPER 2: PRINT AT 2,2;l$( T
O 13);AT 4,2;l$( TO 13);AT 3,2;"
 ";AT 3,14;" ";AT 2,17;l$( TO 13
);AT 4,17;l$( TO 13);AT 17,17;l$
( TO 13)
1110 FOR i=2 TO 17: FOR j=16 TO
29 STEP 13
1120 PRINT AT i,j;" "
1130 NEXT j: NEXT i
1140 FOR i=5 TO 16: PRINT PAPER
7;AT i,17;l$( TO 12): NEXT i
1150 FOR i=3 TO 18 STEP 3: PRINT
 AT i,3; PAPER 7;l$( TO 11): NEX
T i
1160 PRINT AT 3,19; PAPER 6;"MEM
ORIAS";AT 19,18;d$
1170 FOR i=0 TO 9: PRINT AT i+6,
16: PAPER 2; INK 7;i: NEXT i
1190 PAPER 7: RETURN
1490 REM ■ tecla ■
1500 IF INKEY$="" THEN GO TO 150
0
1505 LET a$=INKEY$
1506 BEEP .05,20
1510 IF INKEY$=a$ THEN GO TO 151
0
1520 IF a$<="9" AND a$>="0" THEN
 LET ord=0: RETURN
1530 IF a$>="A" AND a$<="Z" THEN
 LET ord=1: LET a=c(CODE a$-64):
 RETURN
1540 IF CODE a$=13 THEN LET ord=
1: LET a=1600: RETURN
1550 GO TO 1500
1590 REM ■ enter ■
1600 GO SUB 1700: GO SUB 1720: G
O SUB 1760
1610 LET corrpila=0: RETURN
1690 REM ■ push pila ■
1700 FOR i=6 TO 2 STEP -1: LET p
(i)=p(i-1): NEXT i: LET p(1)=0:
RETURN
1710 REM ■ pinta pila ■
1720 FOR i=2 TO 6: LET q$=STR$ p
(i): GO SUB 4000: PRINT AT 3*i,3
;l$( TO 11);AT 3*i,3;q$: NEXT i:
 RETURN
1730 REM ■ pop pila ■
1740 FOR i=2 TO 5: LET p(i)=p(i+
1): NEXT i: LET p(6)=0: RETURN
1750 REM ■ borra display ■
1760 PRINT AT 3,3;l$( TO 11): RE
TURN
1990 REM ■ coge cifra ■
2000 IF corrpila THEN GO SUB 170
0: GO SUB 1720
2010 LET corrpila=1
2020 GO SUB 1500
2030 IF ord THEN GO TO 2070
2040 IF LEN b$>=9+s THEN BEEP 1,
30: GO TO 2020
2050 LET b$=b$+a$: PRINT AT 3,3;
b$
2060 GO TO 2020
2070 IF a$<>"M" AND a$<>"E" THEN
 LET p(1)=VAL b$: RETURN
2080 IF a$="E" THEN GO TO 2160
2090 LET b$=b$+".": PRINT AT 3,3
;b$
2100 GO SUB 1500
2110 IF ord AND a$<>"M" AND a$<>
"E" THEN LET p(1)=VAL b$: RETURN
2120 IF a$="E" THEN GO TO 2160
2130 IF a$="M" OR LEN b$>=10+s T
HEN BEEP 1,30: GO TO 2100
2140 LET b$=b$+a$: PRINT AT 3,3;
b$
2150 GO TO 2100
2160 LET b$=b$+"E": LET q$=b$: G
O SUB 4000: PRINT AT 3,3;q$: LET
 flag=0
2170 GO SUB 1500
2180 IF ord AND flag=0 AND a$="A
" THEN LET b$=b$+"-": LET q$=b$:
 GO SUB 4000: PRINT AT 3,3;q$: L
ET flag=1: GO TO 2170
2190 IF ord THEN GO TO 2170
2210 LET b$=b$+a$: LET q$=b$: GO
 SUB 4000: PRINT AT 3,3;q$
2220 GO SUB 1500: IF ord THEN BE
EP 1,30: GO TO 2220
2230 LET b$=b$+a$: LET q$=b$: GO
 SUB 4000: PRINT AT 3,3;q$
2240 GO SUB 1500: IF NOT ord THE
N BEEP 1,30: GO TO 2240
2250 LET p(1)=VAL b$: RETURN
2290 REM - rutina error -
2300 PRINT AT 3,3;"Error      ":
 BEEP 2,30: GO SUB 1760: LET q$=
STR$ p(1): GO SUB 4000: PRINT AT
 3,3;q$: RETURN
2490 REM ■ operac. binarias ■
2500 IF a$="H" AND p(2)<0 THEN G
O SUB 2300: RETURN
2505 IF a$="H" THEN LET p(1)=p(2
)↑p(1)
2510 IF a$="J" THEN LET p(1)=p(2
)-p(1)
2520 IF a$="K" THEN LET p(1)=p(2
)+p(1)
2525 IF a$="V" AND p(1)=0 THEN G
O SUB 2300: RETURN
2530 IF a$="V" THEN LET p(1)=p(2
)/p(1)
2540 IF a$="B" THEN LET p(1)=p(1
)*p(2)
2560 LET corrpila=1: GO SUB 1760
: LET q$=STR$ p(1): GO SUB 4000:
 PRINT AT 3,3;q$: GO SUB 1740: G
O SUB 1720: RETURN
2590 REM ■ signo ■
2600 LET s=1: LET b$="-": GO SUB
 1760: PRINT AT 3,3;b$
2610 GO SUB 2000
2620 GO SUB a: RETURN
2690 REM ■ pinta memorias ■
2700 FOR i=1 TO 10: LET q$=STR$
m(i): GO SUB 4000: PRINT AT 5+i,
18;l$( TO 11);AT 5+i,18;q$: NEXT
 i: RETURN
2790 REM ■ mete en memoria ■
2800 PRINT AT 5,21; PAPER 0; INK
 7;"STO"
2810 GO SUB 1500: IF ord THEN GO
 TO 2810
2820 LET m(1+VAL a$)=p(1)
2830 GO SUB 2700
2840 PRINT AT 5,21;"   "
2850 RETURN
2890 REM ■ saca de memoria ■
2900 PRINT AT 5,21; PAPER 0; INK
 7;"DTA"
2910 GO SUB 1500: IF ord THEN GO
 TO 2910
2920 IF corrpila THEN GO SUB 170
0: GO SUB 1720
2930 LET corrpila=1
2940 LET p(1)=m(1+VAL a$)
2950 GO SUB 1760: LET q$=STR$ p(
1): GO SUB 4000: PRINT AT 3,3;q$
2960 PRINT AT 5,21;"   "
2970 RETURN
2990 REM ■ clear ■
3000 LET p(1)=0: LET corrpila=0:
 GO SUB 1760: RETURN
3010 REM ■ off ■
3020 INPUT "Estas seguro ?";a$
3030 IF a$="S" THEN RUN 10
3040 RETURN
3090 REM ■ operac. monarias ■
3100 IF a$="Z" AND p(1)>0 THEN L
ET p(1)=LN p(1): GO TO 3300
3110 IF a$="Q" THEN LET p(1)=SIN
 p(1): GO TO 3300
3120 IF a$="E" THEN LET p(1)=TAN
 p(1): GO TO 3300
3130 IF a$="W" THEN LET p(1)=COS
 p(1): GO TO 3300
3140 IF a$="R" THEN LET p(1)=INT
 p(1): GO TO 3300
3150 IF a$=d$(6) AND p(1)>0 THEN
  LET p(1)=.434294481933*LN p(1)
: GO TO 3300
3290 GO SUB 1760: PRINT AT 3,3;"
Error": BEEP 2,30
3300 GO SUB 1760: LET q$=STR$ p(
1): GO SUB 4000: PRINT AT 3,3;q$
: RETURN
4000 REM ■ CONTROL DISPLAY ■
4010 IF LEN q$<=11 THEN RETURN
4020 FOR l=1 TO LEN q$
4030 IF q$(l)="E" THEN GO TO 405
0
4040 NEXT l
4050 LET q$=q$( TO l+10-LEN q$)+
q$(l TO ): RETURN
4100 REM ■ funcion segunda ■
4110 GO SUB 1490: IF NOT ord THE
N GO TO 4110
4120 GO SUB 4200+CODE a$
4130 LET q$=STR$ p(1): GO SUB 40
00: PRINT AT 3,3;l$( TO 11);AT 3
,3;q$
4140 RETURN
4200 REM ■ rutinas segundas ■
4268 RETURN
4269 LET p(1)=ATN p(1)
4271 RETURN
4272 IF p(1)>=0 THEN LET p(1)=SQ
R p(1)
4276 RETURN
4277 GO SUB 1700: GO SUB 1710: L
ET p(1)=PI
4280 RETURN
4281 IF ABS p(1)<=1 THEN LET p(1
)=ASN p(1)
4286 RETURN
4287 IF ABS p(1)<=1 THEN LET p(1
)=ACS p(1): RETURN
4288 LET p(1)=EXP p(1)
4500 RETURN
4600 REM ■ rotacion pila ■
4610 LET v=p(6): GO SUB 1690: LE
T p(1)=v: GO SUB 1710: LET q$=ST
R$ p(1): GO SUB 4000: PRINT AT 3
,3;l$( TO 11);AT 3,3;q$
4620 RETURN
5000 REM ■ ayuda ■
5010 PAPER 6: FOR i=1 TO 20: PRI
NT AT i,1;l$( TO 30): NEXT i
5020 FOR i=1 TO 12: PRINT AT 2+i
,5;n$(i);"=";w$(i): NEXT i
5030 PRINT AT 15,5;"X=CLEAR";AT
16,5;"L=LOG";AT 17,5;"Y=AYUDA";A
T 18,5;"N=NEXT";AT 3,19;"O=OFF";
AT 4,19;"S=STORE";AT 5,19;"D=DAT
A";AT 6,19;"P=FUNC.2";AT 9,19; P
APER 5;"FUNCIONES";AT 10,19;"SEG
UNDAS"
5040 FOR i=13 TO 18: PRINT AT i,
19;n$(i);"=";w$(i): NEXT i
5050 IF INKEY$="" THEN GO TO 505
0
5060 GO SUB 1000: GO SUB 1710: L
ET q$=STR$ p(1): GO SUB 4000: PR
INT AT 3,3;q$: GO SUB 2700
5070 RETURN
```

INTRUDER

Luis G. PARERAS

Spectrum 48 K

```
NOTAS GRAFICAS
--------------------
A B C D E F G H I J K L M N O
```

En una época no determinada de nuestro siglo, una importante misión nos llevará a atravesar las más diversas zonas terrestres, a bordo de un tanque, hasta alcanzar la base enemiga.

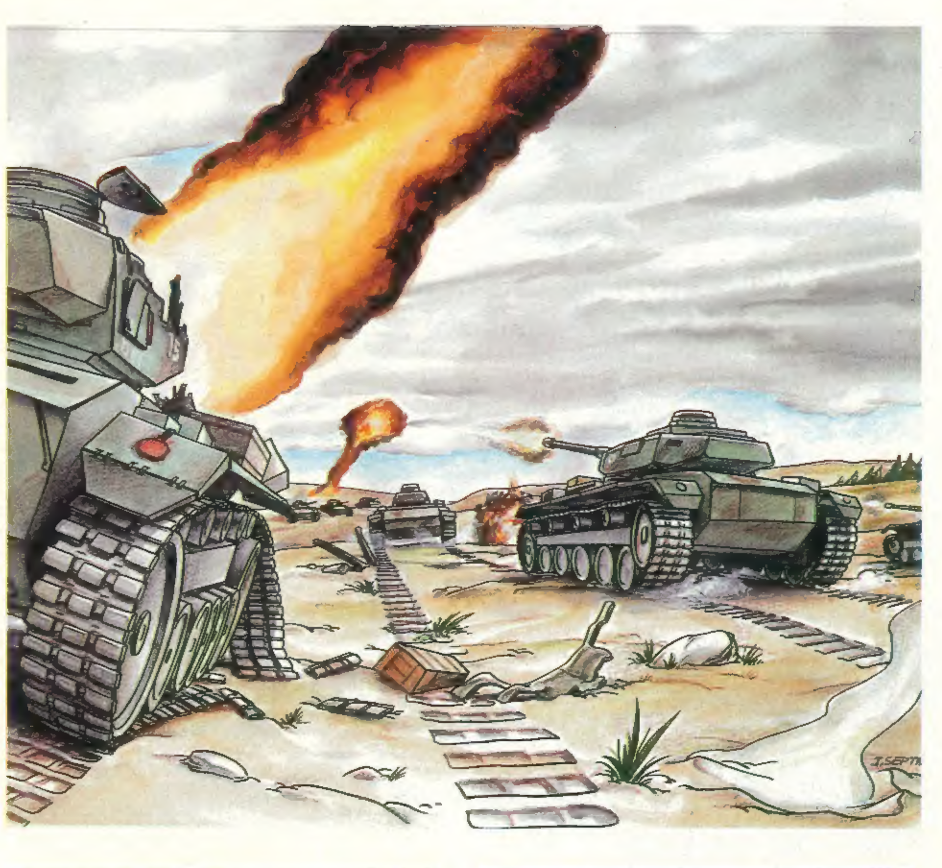
La acción transcurre a lo largo de quince pantallas diferentes, en las que el piloto del tanque de la misión «intruder» (el usuario) tendrá que atravesar múltiples peligros para llegar a la pantalla 15 y concluir su misión. Es fundamental no chocar con árboles, ni con otros tanques y tener cuidado de recoger el fuel que se encuentra en cada pantalla. Para ello, utilizaremos la punta del tanque, pasando por encima del depósito de combustible.

Como consejos importantes, deciros que ninguna pantalla es imposible de superar y debeis tener en cuenta que para destruir un tanque enemigo, hay que disparar a una cierta distancia para conseguir que sea aniquilado por la explosión. En la última pantalla, tendremos que romper el círculo generador de energía de la base enemiga, situado arriba en el centro.

Contamos con tres mandos: «A», derecha. «S», izquierda. «L», fuego.

```
   1 REM ************************
          *     Intruder         *
          *  © Luis G. Pareras   *
          ************************
   2 CLEAR 65159: FOR q=65160 TO
 65356: READ c: POKE q,c: NEXT q
   3 DATA 2,175,8,8,1,1,0,0,0,0,
0,221,42,178,92,221,35,42,123,92
,221,94,5,29,203,35,203,35,203,3
5,22,0,25,235,221,78,0,221,70,1
   4 DATA 221,126,3,245,197,221,
126,2,221,119,6,205,170,34,221,1
19,7,221,52,7,47,230,7,60,221,11
9,8,213,229,205,219,11,225,209
   5 DATA 221,203,4,70,32,5,1,0,
0,24,5,235,70,35,78,235,221,54,9
```

```
,8,221,54,10,9,126,221,53,7,40,9
,7,221,53,10,221,53,7,32,247,203
,33,203,16,23,221,53,10,221,53
   6 DATA 6,32,19,221,53,10,40,6
,7,221,53,10,32,250,119,193,241,
5,61,32,155,201,221,53,9,32,10,2
21,203,4,70,40,4,235,35,78,235
   7 DATA 221,53,8,32,204,221,53
,10,40,6,7,221,53,10,32,250,119,
35,213,229,245,205,219,11,241,22
5,209,221,54,8,8,221,54,7,1,24,1
53
  16 DIM a(4,2)
  17 FOR n=USR "a" TO USR "a"+82
: READ f: POKE n,f: NEXT n
  18 RESTORE 34
  19 FOR n=USR "k" TO USR "k"+39
: READ f: POKE n,f: NEXT n
  20 DATA 1,128,1,128,BIN 001000
01,BIN 10000100,BIN 00111001,BIN
 10011100,BIN 00111111,BIN 11111
100,BIN 00111001,BIN 10011100,BI
N 00111001,BIN 10011100,BIN 0011
1011,BIN 11011100,BIN 00111011,B
IN 11011100
  30 DATA BIN 00111011,BIN 11011
100,BIN 00111011,BIN 11011100,BI
N 00111000,BIN 00011100,BIN 0011
1111,BIN 11111100,BIN 00010000,B
IN 00001000,0,0,0,0
  31 DATA 0,BIN 00011000,BIN 101
00010,BIN 10000110,BIN 00011010,
BIN 01000100,BIN 00111000,0,BIN
01100000,BIN 10001110,BIN 001000
01,BIN 00100100,BIN 10110101,BIN
 10001010,BIN 01100010,0
  32 DATA 0,0,0,0,BIN 00010000,B
IN 00001000,BIN 00111111,BIN 111
11100,BIN 00111000,BIN 00011100,
BIN 00111011,BIN 11011100,BIN 00
111011,BIN 11011100,BIN 00111011
,BIN 11011100,BIN 00111011,BIN 1
1011100
  33 DATA BIN 00111001,BIN 10011
100,BIN 00111001,BIN 10011100,BI
N 00111111,BIN 11111100,BIN 0011
1001,BIN 10011100,BIN 00010001,B
IN 10001000,1,128,1,128,1,128,1,
128
  34 DATA BIN 00001100,BIN 00110
011,BIN 00100000,BIN 01001000,BI
N 01010001,BIN 10100001,BIN 1010
0001,BIN 10100110,BIN 11100000,B
IN 00011000,BIN 00000110,BIN 00
111001,BIN 01000101,1,BIN 000100
10,BIN 00100001
  35 DATA 128,BIN 10010100,BIN 1
0100100,BIN 10100011,BIN 0101000
0,BIN 10001001,BIN 01111010,BIN
00000111,BIN 00100001,BIN 110000
01,BIN 00001001,BIN 01001010,BIN
 00110010,BIN 11000010,BIN 00101
100,BIN 11110000
  36 DATA 24,24,60,60,118,114,11
4,60
  40 POKE 65162,16: POKE 65163,1
6: POKE 65165,1
 400 LET pun=0: LET fu=100: LET
mu=40: LET ti=0: LET po=0
 420 GO SUB 9900
 430 LET ty=174
 431 LET x=79: LET y=18
 435 LET ty2=174
 436 LET po=po+6.7
 450 LET muerto1=0: LET muerto2=
0
 499 POKE 65164,1
 500 IF INKEY$="a" THEN LET x=x-
2
 505 IF INKEY$="0" THEN GO SUB 4
000
 510 IF INKEY$="s" THEN LET x=x+
2
 515 IF INKEY$="l" AND y<105 AND
 mu>0 THEN LET fu=fu-1: LET mu=m
u-1: GO SUB 9000
 520 LET y=y+2
 521 IF y>173 THEN GO SUB 9900:
GO TO 430
 523 GO SUB 5000
 524 PRINT AT 2,22;INT fu;" "
 525 LET fu=fu-.7: IF fu<1 THEN
PRINT "No tienes fuel.": PRINT "
Eres un inepto": PRINT "El alto
mando nunca": PRINT "Debio encom
endarte esta": PRINT "mision.":
STOP
 526 LET ti=ti+.2
 527 PRINT AT 8,22;INT ti
 530 POKE 65160,x: POKE 65161,y
 535 INK 9
 540 RANDOMIZE USR 65171
 545 GO SUB 7500
 550 GO SUB 7000
 600 GO TO 500
4000 REM *** Demo mode ****
4010 FOR g=1 TO 15: RESTORE 9959
+g: GO SUB 9800: GO SUB 9900: PA
USE 50: NEXT g
4020 RETURN
5000 REM *** Cogida de fuel
5010 LET fy=(21-f1)*8+1: LET fx=
f2*8+4
5020 IF (y+2=fy OR y+3=fy) AND x
+8>fx-3 AND x+8<fx+3 THEN LET fu
=fu+60: PRINT AT f1,f2;" "
5100 RETURN
6000 REM *** Ultima pantalla
6001 PLOT 0,0: DRAW 175,0: DRAW
0,175: DRAW -175,0: DRAW 0,-175
6010 PLOT 73,0: DRAW 0,40: DRAW
-50,40: DRAW 0,40: DRAW 50,40: D
RAW 30,0,-3
6020 DRAW 50,-40: DRAW 0,-40: DR
AW -50,-40: DRAW 0,-40: CIRCLE 8
7,168,5
6100 RETURN
7000 REM *** Tanques enemigos
7010 POKE 65165,7
```



```
7022 POKE 65160,pos
7025 POKE 65161,ty
7030 IF muerto1=0 THEN RANDOMIZE
 USR 65171
7035 LET ty=ty-2: IF ty<18 THEN
LET muerto1=1: LET ty=ty+1
7040 IF ty2<18 THEN LET muerto2=
1: LET ty2=ty2+1
7041 IF ty<120 THEN POKE 65160,p
os2: POKE 65161,ty2: LET ty2=ty2
-2: IF muerto2=0 THEN RANDOMIZE
USR 65171
7050 POKE 65165,1
7200 RETURN
7500 REM *** Colisiones ***
7510 IF POINT (x+8,y+1)=1 OR POI
NT (x-1,y+1)=1 OR POINT (x-1,y-1
7)=1 OR POINT (x+17,y+1)=1 OR PO
INT (x+17,y-17)=1 THEN BEEP .01,
-20: GO SUB 7700
7600 RETURN
7700 REM **** Explosion tanque
7710 LET ey=21-(y/8): LET ex=x/8
7720 PRINT AT ey,ex;"[illegible]";AT ey+
1,ex;"[illegible]";AT ey+2,ex;"[illegible]"
7730 FOR n=1 TO 10: BEEP .005,n*
2: NEXT n
7740 PRINT AT ey,ex;"[illegible]";AT ey+
1,ex;"[illegible]";AT ey+2,ex;"[illegible]"
7750 BEEP 1,10: PRINT AT ey,ex;"
   ";AT ey+1,ex;"   ";AT ey+2,ex
;"   ": BEEP 1,20
7780 STOP
7799 RETURN
9000 REM ****** Disparo ******
9001 PRINT AT 5,22;mu
9005 PLOT x+8,y
9007 FOR s=1 TO 2
9010 FOR a=1 TO 70 STEP 6
9020 OVER 1: PLOT x+8,y+a: DRAW
INK 0;0,3
9030 NEXT a
9032 PLOT x+8,y: DRAW OVER 1;0,5
9035 NEXT s
9040 OVER 0
9045 LET cy=21-((y+70)/8): LET c
x=x/8
9050 PRINT AT cy,cx;"[illegible]"
9055 IF col=0 AND y>92 AND y<100
 THEN LET y=173: RESTORE 9960
9060 BEEP .01,20: BEEP .01,10: B
EEP .05,0: PRINT AT cy,cx;"[illegible]":
BEEP .05,0: PRINT AT cy,cx;" "
9070 IF x>pos-5 AND x<pos+5 AND
ty-y>73 AND ty-y<87 THEN LET mue
rto1=1: LET pun=pun+100: PRINT A
T 14,22;pun: POKE 65160,pos: POK
E 65161,ty: POKE 65164,0: RANDOM
IZE USR 65171: POKE 65164,1
9080 IF x>pos2-5 AND x<pos2+5 AN
D ty2-y>73 AND ty2-y<87 THEN LET
 muerto2=1: LET pun=pun+100: PRI
NT AT 14,22;pun: POKE 65160,pos2
: POKE 65161,ty2: POKE 65164,0:
RANDOMIZE USR 65171: POKE 65164,
1
9100 RETURN
9800 REM *** Indicadores ***
9805 PAPER 0: INK in: BORDER 0:
CLS
9810 PRINT AT 1,22; INK 2;"Fuel:
 ";AT 4,22; INK 4;"Municion: ";A
T 7,22; INK 5;"Tiempo: ";AT 10,2
2; INK 3;"Porcentaje: ";AT 13,22;
 INK 6;"Puntuacion:"
9820 INK 7: PRINT AT 2,22;INT fu
;AT 5,22;INT mu;AT 8,22;INT ti;A
T 11,22;INT po;AT 14,22;pun;AT 1
6,22; INK 4;"Zona: ";AT 17,22; P
APER col; INK 9;a$
9830 INK in
9899 RETURN
9900 REM *** Crea Pantallas ***
9920 READ f1,f2,a$,col,in,pos,po
s2
9921 GO SUB 9800: IF col<>0 THEN
 GO SUB 9990
9922 IF col=0 THEN GO SUB 6000
9924 PRINT AT f1,f2; INK 9;"[illegible]"
9925 FOR w=1 TO 4: READ a(w,1),a
(w,2): PRINT AT a(w,1),a(w,2); I
NK 9;"[illegible]";AT a(w,1)+1,a(w,2); IN
K 9;"[illegible]": NEXT w
9960 DATA 5,9,"Desierto 1",6,2,5
0,85,4,4,10,16,16,13,12,4
9961 DATA 2,16,"Desierto 2",6,1,
105,60,6,4,12,5,10,16,18,4
9962 DATA 4,6,"Tropico 1",4,2,10
0,70,10,5,18,4,5,15,8,16
9963 DATA 1,5,"Tropico 2",4,1,88
,110,11,5,13,6,15,7,5,16
9964 DATA 9,6,"Helada 1",5,1,60,
100,9,15,13,16,5,5,18,16
9965 DATA 9,5,"Helada 2",5,7,60,
90,11,15,5,5,5,16,19,5
9966 DATA 21,30,"Inferno 1",2,6,
90,70,3,5,5,16,12,6,15,14
9967 DATA 6,11,"Inferno 2",2,7,6
7,120,2,11,6,5,14,12,19,6
9968 DATA 10,6,"Atlantis 1",1,4,
60,80,2,14,6,16,13,13,19,5
9969 DATA 5,14,"Atlantis 2",1,2,
50,90,4,16,10,14,16,9,14,15
9970 DATA 2,8,"Intruder 1",2,6,9
0,75,4,5,4,16,9,7,9,14
9971 DATA 3,9,"Intruder 2",2,0,9
0,90,5,5,5,8,16,15,5,14
9972 DATA 3,8,"Alien 1    ",7,2,1
10,115,4,5,6,8,8,11,8,17
9973 DATA 8,8,"Alien 2    ",7,3,8
0,80,2,7,2,13,8,5,8,15
9980 DATA 0,0,"Alien Base",0,6,1
,155,4,10,6,10,11,6,11,13
9989 RETURN
9990 REM **** Montanas ****
9991 FOR n=0 TO 21: PRINT AT n,0
; PAPER col;"
   ": NEXT n: PLOT 0,0: DRAW 175
,0: DRAW 0,175: DRAW -175,0: DRA
W 0,-175
9992 LET lo=20
9993 FOR n=0 TO 175
9994 LET lo=lo+(RND*2)-1: IF lo<
2 THEN LET lo=2
9995 PLOT 0,n: DRAW lo,0
9996 PLOT 175,n: DRAW -lo,0
9997 NEXT n
9998 PAPER col: INK 0
9999 RETURN
```

FAVORITOS SPECTRUM

ZAFI CHIP

1985

MATCH DAY



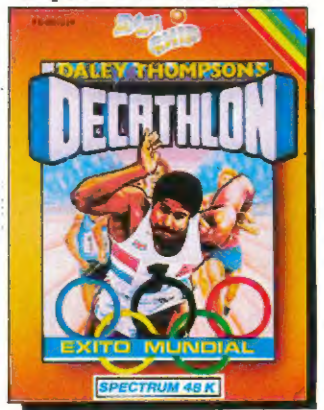
La emoción

HUNCHBACK II
Disponible en Commodore



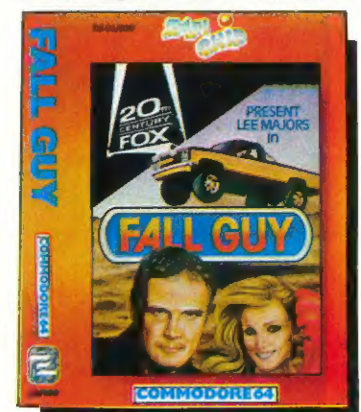
El rescate

DECATHLON
Disponible en Commodore



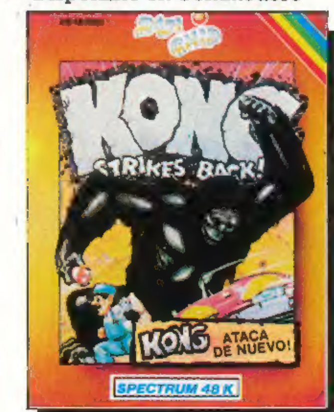
La victoria

DUKES OF HAZZARD
Pronto disponible en Commodore



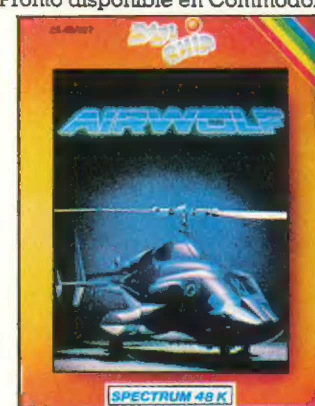
La aventura

KONG STRIKES BACK
Disponible en Commodore



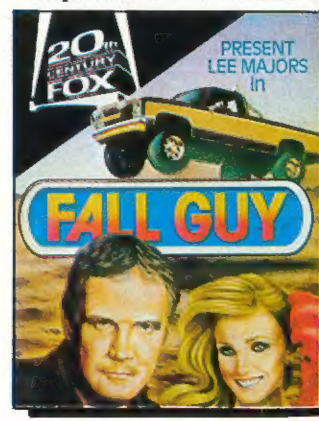
La furia

AIRWOLF
Pronto disponible en Commodore



El riesgo

FALL GUY
Disponible en Commodore



La acción



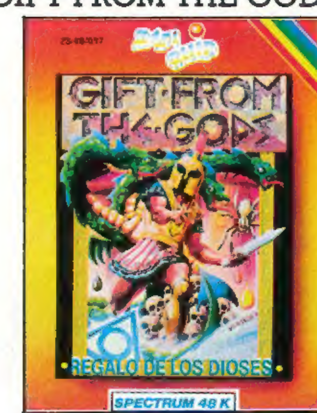
ZAFI CHIP te presenta los superventas en todo el mundo.
La más completa gama de juegos para tu Sinclair Spectrum 48 K. ¡Disfrútalos!

EVERYONE'S A WALLY
Pronto disponible en Commodore



La diversión

GIFT FROM THE GODS



El destino

¡Toda la diversión a tu alcance!

INSTRUCCIONES EN CASTELLANO

SI ESTAN AGOTADOS EN TU TIENDA HABITUAL ¡¡LLAMANOS!!

ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141 - 28046 Madrid.
Tel. 459 30 04. Telex: 22690 ZAFIR E / Tel. Barcelona 209 33 65

TIR NA NOG



Gargoyle Games / Software Center

48K

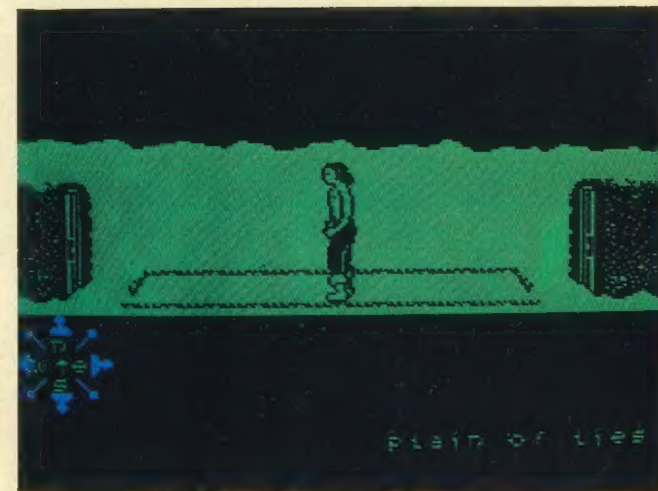
Tipo de juego: Videoaventura

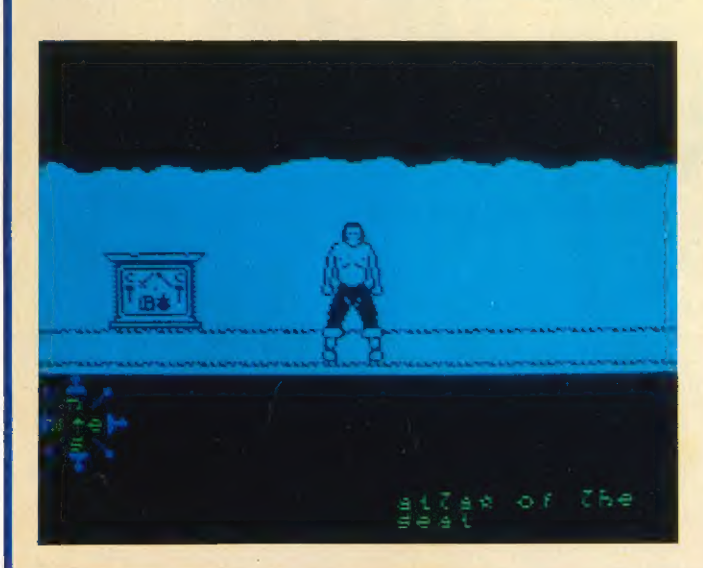
P.V.P.: 2.700

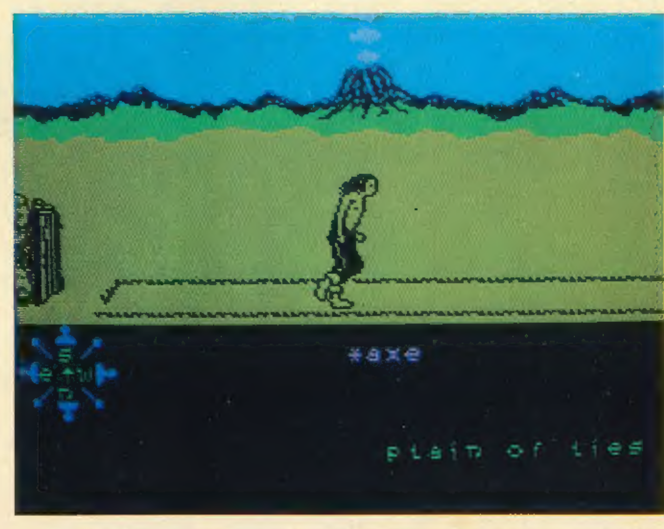
Tir Na Nog es un juego en el que los gráficos han sido cuidados al máximo, y el movimiento tratado igual que si fuera una película.
El personaje central tiene 56 pixels de alto, y se controla a través del teclado. Han sido necesarias un total de 64 estructuras para conseguir la animación completa de la figura.

El juego trata de las hazañas del gran héroe Cuchulain, en su entrada a Tir Na Nog. Tiene que encontrar los fragmentos del Sello de Calum. El escenario está situado en los dominios mágicos de la mitología céltica.
La acción se presenta como si una cámara estuviera filmando al personaje principal, y al fondo, en el horizonte, aparece un paisaje que se mueve con nosotros y que cambia dependiendo de cuál sea nuestra posición.
Nuestro héroe puede llevar consigo hasta un máximo de cuatro objetos que pueden coger o soltar cuando quiera. Si entra en combate con algún otro personaje del juego, podrá golpear con cualquiera de los objetos que lleve en ese momento. Existe en la parte inferior de la imagen una zona destinada a enviarnos mensajes de información, y nos sitúa, además, en el plano correcto donde nos encontramos, de modo que podamos orientarnos.
En el teclado encontramos todas las posibilidades necesarias para manejar al personaje, coger objetos, golpear con éstos, ver al personaje desde diferentes ángulos, según nuestra propia elección, dejar el objeto previa elección del mismo y una larga serie de funciones que serán decisivas para lograr superar con éxito el juego, como poder pararlo, volver al menú, grabarlo y cargarlo, etc.
Para jugar a este juego es necesario seguir una estrategia que va a ser decisiva a la hora de intentar conseguir el objetivo del mismo, por eso, es necesario tener muchos aspectos en cuenta, como es el caso de los que vamos a citaros:



















Exploración. Es necesario encontrar el camino que vamos a seguir, de entre la cantidad de ellos que se cruzarán ante nosotros.
Interacción. Encontraremos muchos personajes a lo largo del juego, algunos de los cuales podrán ayudarnos si sabemos cómo dirigirnos hacia ellos, mientras que contra otros tendremos que luchar para poder proseguir nuestro camino.

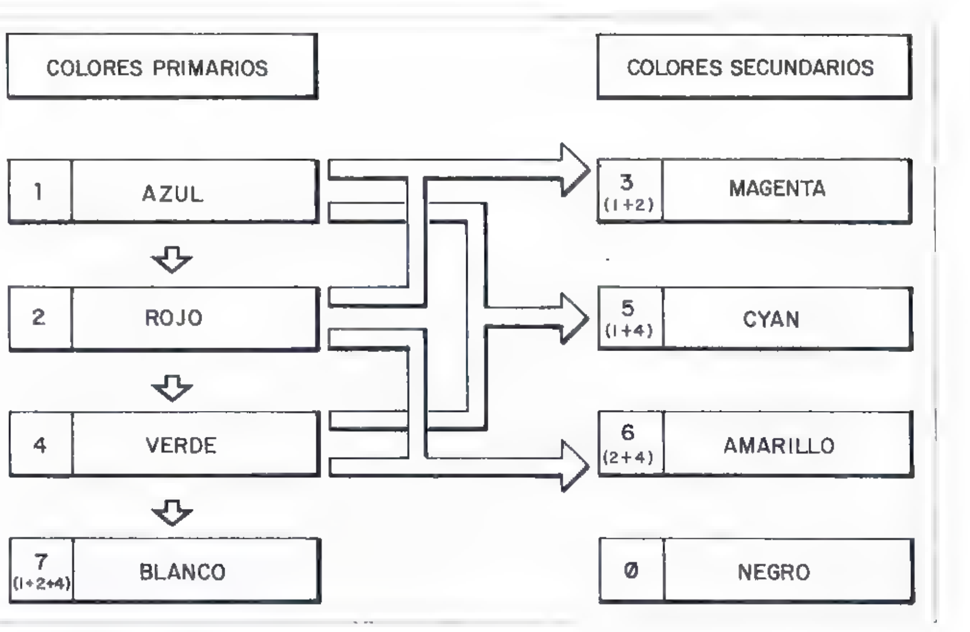
Búsqueda. Es el objetivo principal del juego, pero deberemos en muchas ocasiones buscar otras cosas secundarias según se desarrolle el juego.

Generalidades. Como en la vida real, muchas de las situaciones dependerán de lo que hayamos hecho anteriormente. Hay puertas, diferentes rutas, armas, tesoros y diversos objetos para recoger.

Valoración. Es una aventura con gráficos totalmente en movimiento. El personaje central es una maravilla que se desplaza por la pantalla, dándonos la sensación real de que está andando de verdad. La idea es muy original y el juego bastante difícil.

Nos puede llevar días y días conseguir acabar la aventura, e incluso si lo consiguiéramos, al volver a jugar las situaciones podrían ser bastantes distintas. El juego viene acompañado de un fragmento del Lebhar Bloadhach, que lleva la historia del Sello de Callum y de la caída de Shide.

Originalidad	* * * *
Gráficos	* * * *
Movimiento	* * * * *
Sonido	
Valoración	* * * * *



Síntesis aditiva de colores.

al ejecutarse.

Tres cosas que facilitan bastante la depuración son:

- DIAGRAMAS DE FLUJO
- LISTADOS
- USO DE SUBRUTINAS

El tener confeccionado el diagrama de flujos nos permite ver las funciones que realiza el programa paso a paso y los diversos caminos que debe seguir según se cumplan ciertas condiciones. Podemos localizar, con su ayuda, la zona o el bloque de instrucciones donde falla el programa.

Si se dispone de una impresora, podemos obtener un listado y así localizar con mayor facilidad las sentencias que consideremos oportunas.

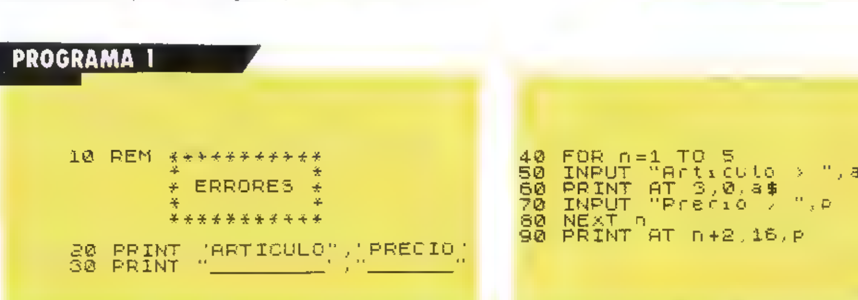
El uso de subrutinas facilita la depuración, ya que éstas pueden ser comprobadas individualmente, de esta manera se pueden tener ciertas garantías de que funcionan cuando se depura el programa final.

Es importante tener localizada la zona a partir de la cual se desea comenzar la depuración, por ejemplo, en el siguiente programa, que tiene dos errores, empezaríamos a depurar a partir de la línea 4Ø, ya que si lo ejecuta observará que se visualizan las constantes de cadena de las líneas 2Ø y 3Ø.

Para corregir el ejemplo, deberá modificar la línea 6Ø

```
6Ø PRINT AT n + 2,Ø, a$
```

PROGRAMA 1



```
10 REM ***********
       *         *
       * ERRORES *
       *         *
       ***********
20 PRINT "ARTICULO","PRECIO"
30 PRINT "__________","______"
40 FOR n=1 TO 5
50 INPUT "Articulo ? ",a$
60 PRINT AT 3,0,a$
70 INPUT "Precio ? ",p
80 NEXT n
90 PRINT AT n+2,16,p
```

Rutina «cortar» el mazo.

Programa

El programa número «2» permite jugar con el ordenador, al conocido juego de cartas: «Las Siete y Media».

El ordenador hace el papel de «banca» y por tanto, baraja y reparte las cartas; nosotros tenemos la posibilidad de indicar por qué número deseamos «cortar» la baraja.

Después de repartir las dos primeras cartas, una para él y otra para nosotros, nos irá preguntando sucesivamente si deseamos otra carta o no.

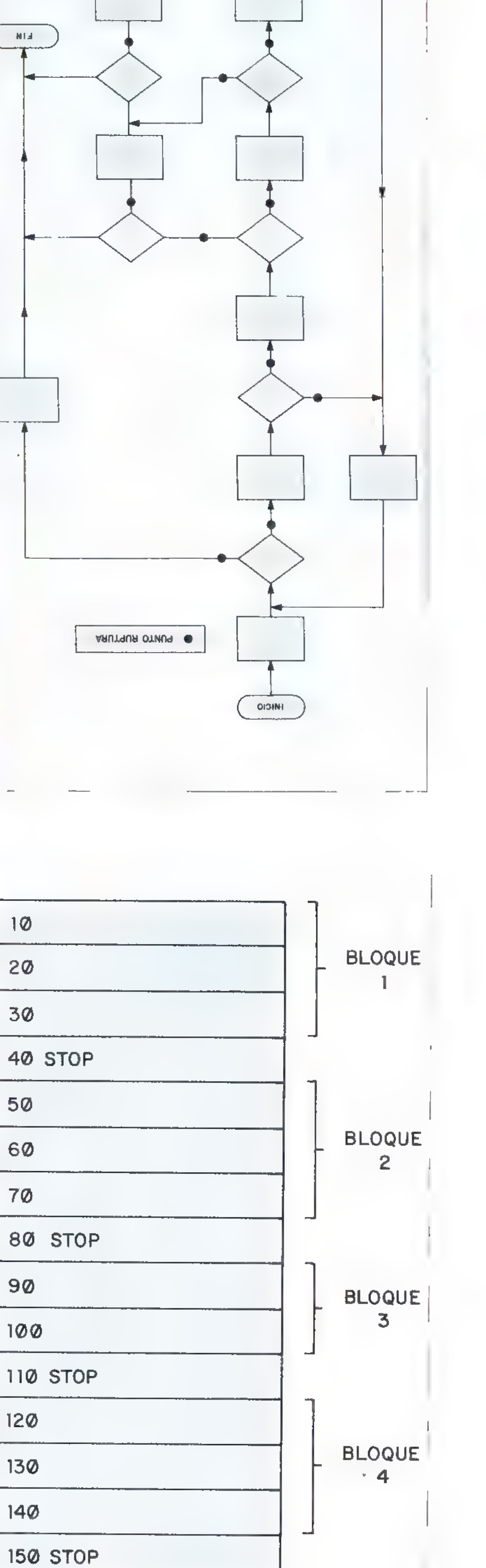
Al final de la partida se presentan los tantos obtenidos por cada uno de los dos jugadores y si deseamos jugar otra «mano».

Para dar mayor emoción a la partida, al principio de ésta, tenemos un crédito de «10.000» ptas. Las apuestas son de «100» ptas., ganamos el doble si obtenemos «siete y media» y el ordenador no, ya que en igualdad de puntos gana la banca.

¡BUENA SUERTE!

Hay una serie de matrices utilizadas en la confección del programa, que se encuentran dimensionadas en las líneas 9Ø a 125.

La matriz «T$» almacena el nombre de cada carta (as de oros, dos de ...., etc.). Es una tabla bidimensional de «1Ø» por «4», ya que la baraja consta de 4 «palos» de diez cartas. La tercera dimensión (2Ø) indica el número máximo de caracteres de cada elemento, y se encuentra ligeramente *sobredimensionada* ya que con dieciocho hubiera bastado, por que el contenido de mayor longitud va a ser «caballo de espadas» (16 letras + 2 espacios).



1Ø	BLOQUE 1
2Ø	
3Ø	
4Ø STOP	
5Ø	BLOQUE 2
6Ø	
7Ø	
8Ø STOP	
9Ø	BLOQUE 3
1ØØ	
11Ø STOP	
12Ø	BLOQUE 4
13Ø	
14Ø	
15Ø STOP	

«Depurado por partes».

Los blucles anidados «p» y «n» (líneas 14Ø-2ØØ y 16Ø-19Ø) son los encargados de asignar a la matriz «T$» los valores de cadena especificados en las sentencias «DATA» (líneas 251 a 26Ø).

En los juegos de naipes, cada carta tiene un valor relativo distinto; en el caso de las siete y media, las cartas del «as» al «siete» tienen un valor idéntico al que figura en una de sus esquinas. Por el contrario, las «figuras» (sota, caballo y rey) tienen un valor de medio punto. Con estos requisitos, los blucles 31ø-33ø y 34Ø-36Ø asignan a los elementos de la matriz «V» el valor correspondiente.

«V» es una matriz unidimensional de diez elementos, uno por cada valor de carta. La relación entre el subíndice y la carta es el siguiente:

```
1 - AS
2   DOS
3   TRES
4 - CUATRO
5   CINCO
6   SEIS
7 - SIETE
8   OCHO
9 - CABALLO
10  REY
```

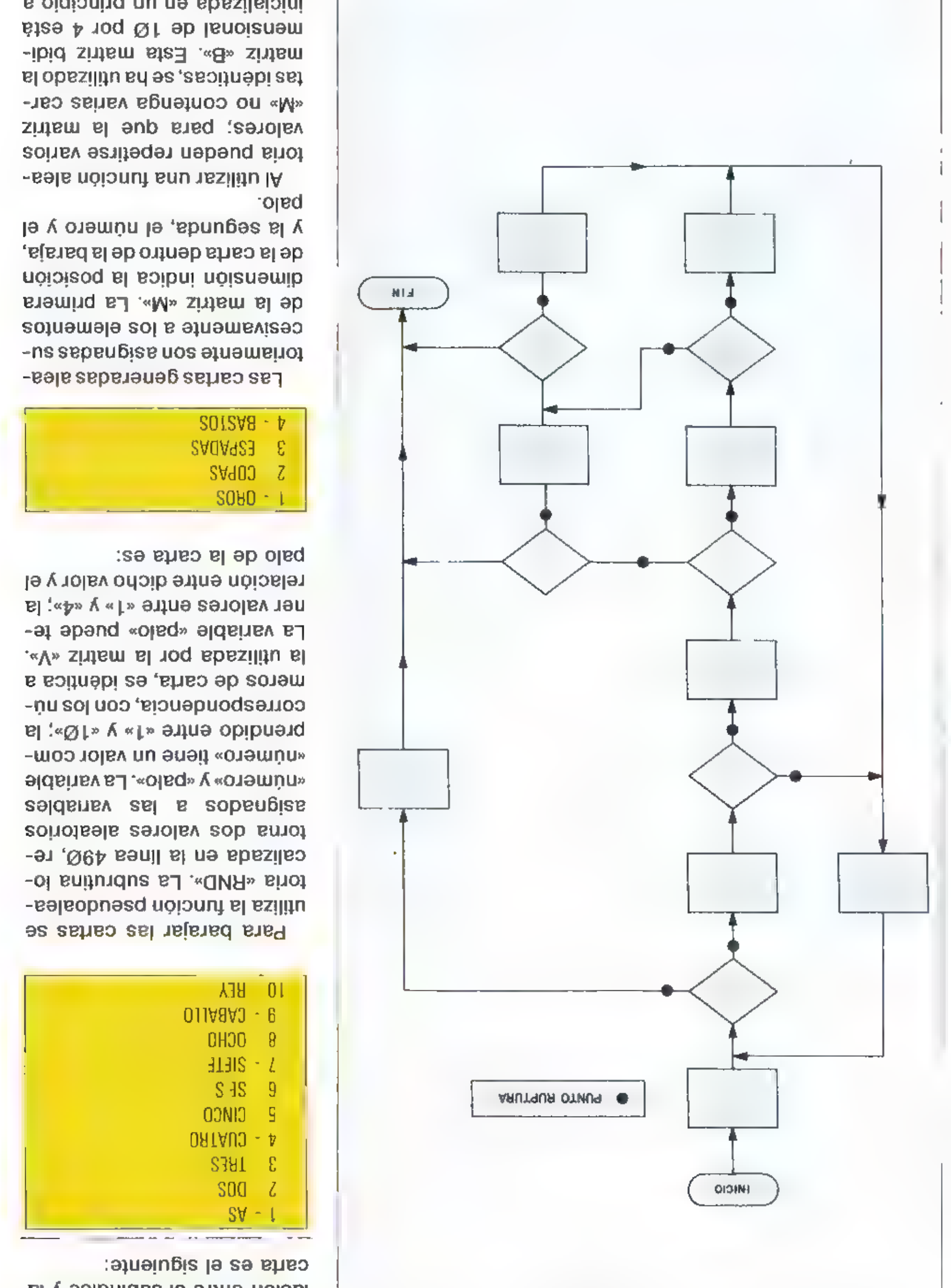
Para barajar las cartas se utiliza la función pseudoaleatoria «RND». La subrutina localizada en la línea 49Ø, retorna dos valores aleatorios asignados a las variables «número» y «palo». La variable «número» tiene un valor comprendido entre «1» y «1Ø»; la correspondencia, con los números de carta, es idéntica a la utilizada por la matriz «V». La variable «palo» puede tener valores entre «1» y «4»; la relación entre dicho valor y el palo de la carta es:

```
1 - OROS
2   COPAS
3   ESPADAS
4 - BASTOS
```

Las cartas generadas aleatoriamente son asignadas sucesivamente a los elementos de la matriz «M». La primera dimensión indica la posición de la carta dentro de la baraja, y la segunda, el número y el palo.

Al utilizar una función aleatoria pueden repetirse varios valores; para que la matriz «M» no contenga varias cartas idénticas, se ha utilizado la matriz «B». Esta matriz bidimensional de 1Ø por 4 está inicializada en un principio a cero.

Los valores proporciona-



Puntos estratégicos de ruptura.

DEPURACION DE PROGRAMAS

En capítulos anteriores se ha ido explicando la mayor parte del juego de sentencias del Spectrum, así como una serie de conceptos:Bucles, Subrutinas,Funciones, Matrices, etc. Con todo este *material* el iniciado en programación puede ya confeccionar un volumen importante de programas; por este motivo es necesario que conozca una serie de técnicas destinadas a la *depuración* o puesta a punto de programas; también son conocidas por el término inglés *debugging*.

BORDE (BORDER)
CARACTER O TINTA (INK)
FONDO (PAPER)
BORDER 1: PAPER 2: INK Ø : CLS

Zonas de la pantalla.

Errores

Es muy difícil que un programa, medianamente complicado, funcione a la primera, ya que hay muchos factores que deben tenerse en cuenta.

Los tipos de error que surgen al elaborar un programa,pueden clasificarse, principalmente, en tres apartados:

- SINTACTICOS
- DE DEFINICION
- DE ESTRUCTURA

Los errores *sintácticos* son aquellos que se producen al teclear un argumento no compatible con una sentencia determinada, por ejemplo:

```
- NEW 32
- LET "A$" = "abc"
- CLS fin
- PRINT $
```

Este tipo de errores son detectados inmediatamente por el intérprete BASIC, al pulsar «ENTER». Una interrogación parpadeante se posiciona en las proximidades del error, indicando que no entiende dicha sentencia.

Revise la línea introducida; si no encuentra el fallo, consulte, en el capítulo correspondiente del libro, las posibles estructuras que puede adoptar dicha sentencia.

Como puede observar, los errores sintácticos son detectados y corregidos en la fase de edición de programas, a diferencia de otros ordenadores en los que se realiza posteriormente.

Otro tipo de errores se manifiestan al ejecutarse el programa, presentando el correspondiente mensaje de error. Estos errores son originados principalmente por no haber definido previamente una variable, por utilizar un subíndice fuera de rango, por utilizar un parámetro erróneo en una función, etc.

El mensaje presentado sirve únicamente de referencia para encontrar el error, ya que indica el tipo y la línea en que lo ha detectado (ver pag. 96). El número de línea no siempre significa que el error se encuentre en la misma, sino que está relacionado con ella, por ejemplo, un error con mensaje:

```
2 Variable not found, 7Ø-1
```

suponiendo que la línea 7Ø fuera:

```
7Ø PRINT mes
```

significa que el argumento de «PRINT» al no ir entrecomillado, lo interpreta como variable y no encuentra su asignación inicial. Si efectivamente «mes» era el nombre de una variable, solucionaremos el problema asignando a esta un valor en la zona del programa que corresponda, por ejemplo:

```
25 LET mes = 8
```

Si «mes» no fuera una variable, sino que por el contrario fuera una cadena a visualizar, el error se encontraría en la misma línea 7Ø, ya que debería ser:

```
7Ø PRINT «mes»
```

Los errores más difíciles de localizar son aquellos que no generan ningún tipo de mensaje de error, pero que hacen que nuestro programa no realice las funciones que en un principio teníamos previstas. Al corregir este tipo de errores, de *estructura*, están concebidas las técnicas de depuración.

AZUL (1)
MAGENTA (3)
ROJO (2)
AMARILLO (6)
VERDE (4)
CYAN (5)
BLANCO (7)

Estrella de colores.

Depuración

La tarea de depurar no es muy complicada, pero sí laboriosa, aunque a veces basta con echar una ojeada al *listado*, para descubrir el error.

La complejidad de la depuración depende del número de bifurcaciones que tenga el programa, ya que puede ser bastante elevado el número de caminos distintos que siga

dos por la subrutina de generación de cartas aleatorias son utilizados como subíndices de la matriz «B»; si el elemento direccionado tine valor «Ø», es que la carta no se ha generado anteriormente; si, por el contrario, tiene valor «1», indica que la carta se repite. En el primer caso, los valores de las variables «número» y «palo» se asignan al elemento correspondiente de la matriz «M» y el valor «1» al elemento direccionado de la matriz «B». En el segundo caso, no se asigna ningún valor a la matriz «M» y el programa genera dos nuevos valores que vuelven a ser conprobados.

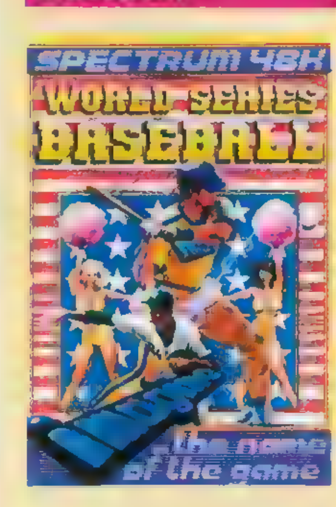
La rutina encargada de «cortar» la baraja se encuentra localizada en las líneas 61Ø a 675. En la matriz «C» quedan las cartas ordenadas y listas para poder comenzar la partida.

La transferencia de cartas entre la matriz «M» y «C» se realiza en dos fases. En la primera, se transfieren los elementos situados en las primeras posiciones, es decir, desde el número uno hasta el direccionado por la variable «número»; en la segunda, se realiza la transferencia de las restantes. En la matriz «C» estos dos bloques quedan intercambiados.

La estructura general del programa es:

1Ø	Comentario con el nombre del programa.
15	Asignación del color azul para el borde, verde para el fondo y negro para los caracteres
2Ø-26	Carátula del programa
4Ø 75	Asignación de valores a las variables «crédito» y «manos». Esta última contabiliza el número de partidas jugadas.
9Ø-125	Dimensionado de las matrices explicadas anteriormente.
14Ø 2ØØ	Bloque de asignación de los nombres de las cartas a la matriz «T$»
251 26Ø	Datos relativos a los nombres de las cartas
31Ø 36Ø	Blucles para la asignación de valores.
365	Llamada a la subrutina de visualización de instrucciones
372	Inicialización de la función aleatoria
38Ø-4ØØ	Mensaje de espera.
41Ø 47Ø	Asignación aleatoria de valores, de carta, a la matriz «M»
49Ø 52Ø	Generación aleatoria de «número» y «palo»
54Ø 57Ø	Introducción del número de carta por donde se desea cortar la baraja.
572	Comprobación de que el valor introducido no es una cadena vacía.
574 578	Bucle para detectar si hay algún valor no numérico dentro de la variable «N$».
58Ø	Evaluación de la cadena «N$»
59Ø	Comprobación de que el valor está dentro de los márgenes «1» a «4Ø»
592-594	Mensaje de espera.
6ØØ	Cálculo del número de cartas que hay desde la «cortada» hasta el final
61Ø 675	Transferencia de valores entre las matrices «M» y «C» (cortar la baraja)
68Ø 69Ø	Pantalla de «ánimo»
7ØØ 715	Temporización hasta que se pulsa una tecla y borrado de la pantalla
722	Invitación a jugar.
73Ø	Inicialización de los tantos del jugador
74Ø	Asignación de la primera carta del jugador
76Ø	Llamada a la subrutina de presentación de cartas.
77Ø	Asignación de los tantos del jugador
78Ø	Asignación de la segunda carta al ordenador
8ØØ-85Ø	Presentación de la carta y asignación de su valor
87Ø	Invitación a pedir otra carta.
88Ø-92Ø	Comprobación de la respuesta
93Ø	Borrar el mensaje de la zona inferior.
932	Temporización entre dos jugadas
934	Asignación de la siguiente carta al jugador.
95Ø	Llamada a la subrutina de presentación de cartas
96Ø	Incrementar los tantos del jugador
97Ø	Comprueba si el jugador se ha «pasado»
98Ø	Comprueba si el jugador tiene «7 1/2».
99Ø	Salto al mensaje de invitación
1ØØØ 1Ø2Ø	Mensaje de presentación de los tantos obtenidos por el jugador, en el caso de «pasarse»
1Ø3Ø	Asignación a la variable «ganador» del valor (Ø), este indica que el ordenador gana.
1Ø35	Inicialización de los tantos del ordenador, para poder visualizar «?».
1Ø4Ø 1Ø5Ø	Temporización hasta que se pulsa una tecla.
1Ø8Ø 1Ø9Ø	Mensaje de obtención, por parte del jugador, de «7 1/2»
11ØØ	Indica cuál es la siguiente carta que debe «pedir» el ordenador
111Ø 112Ø	Temporiza hasta que se pulsa una tecla.
115Ø 117Ø	Mensaje de visualización de los puntos obtenidos por el jugador en el caso de «quedarse»
1175	Borra el mensaje de la zona inferior.
118Ø 119Ø	Temporización hasta que se pulsa una tecla.
122Ø	Inicialización de los tantos del ordenador
123Ø	Asigna la segunda carta al ordenador.
124Ø	Indicación de que juega el ordenador
1252	Coordenada «y» de la siguiente carta a visualizar
126Ø	Llamada a la subrutina de presentación de cartas
127Ø	Asignación de los tantos del ordenador
128Ø	Asignación de la siguiente carta a escoger
13ØØ	Comprueba si el ordenador se ha «pasado»
131Ø	Comprueba si el ordenador ha obtenido «7 1/2»
132Ø	Comprueba si no se ha pasado
133Ø	Asignación del valor «1» a la variable «ganador». El ordenador pierde
134Ø 136Ø	Mensaje de presentación de los tantos del ordenador, en el caso de «pasarse»
137Ø 138Ø	Temporiza hasta que se pulsa una tecla.
141Ø	Asignación del valor (Ø) a la variable «ganador». El ordenador gana
142Ø	Mensaje de visualización «7 1/2», por parte del ordenador
143Ø 1432	Temporiza hasta que se pulsa una tecla.
146Ø	Comprueba si el jugador ha obtenido «7 1/2», para seguir pidiendo cartas hasta que gane o se pase
147Ø	Si el jugador se ha «quedado», pide cartas mientras que él tenga menos de seis puntos.
149Ø	Mensaje pidiendo otra carta.
1492	Duración del mensaje.
1494	Borrado del mensaje
15ØØ	Temporización entre dos jugadas
151Ø	Asignación de la siguiente carta
1522	Incrementar, en dos, la siguiente posición
153Ø	Llamada a la subrutina de presentación de cartas
154Ø	Incrementar los tantos del ordenador
157Ø 159Ø	Mensaje de presentación de los tantos del ordenador, en el caso de «quedarse»
1592-1594	Temporiza hasta que se pulsa una tecla
161Ø	Si el ordenador tiene más puntos o igual que el jugador, el ordenador gana. Si tiene menos, el ordenador pierde.
162Ø	
164Ø	Mensaje de que el ordenador gana
165Ø	Mensaje de que el jugador gana
166Ø	Visualización de los tantos del ordenador.
168Ø	Si el jugador se «pasó», el ordenador no indica cuál era su carta
169Ø	Visualización de los tantos del ordenador
17ØØ 171Ø	Temporización hasta que se pulsa una tecla.
174Ø	Si el jugador gana y tiene «7 1/2», la apuesta se duplica
176Ø-177Ø	Cálculo de los nuevos créditos si gana el ordenador
179Ø 18ØØ	Idem. si gana el ordenador
182Ø	Visualización de las «manos» jugadas hasta ese momento
183Ø	Visualización del nuevo crédito del jugador
184Ø	Idem del ordenador
185Ø	Comprueba si el ordenador se ha quedado sin crédito
186Ø	Idem. del ordenador
188Ø	Mensaje de invitación a jugar otra mano
189Ø 193Ø	Comprobación de la respuesta elegida
194Ø	Borrado del mensaje de la zona inferior
1945	Incremento de las «manos» jugadas.
195Ø 199Ø	Inicialización a cero de los elementos de la matriz «B», permite volver a barajar las cartas
2ØØØ	Salto a la rutina de barajear
21ØØ	Comienzo de la subrutina «INSTRUCCIONES»
211Ø	Invitación a visualizar las instrucciones
212Ø-216Ø	Comprobación de la opción elegida
217Ø-221Ø	Visualización de las instrucciones
222Ø 223Ø	Pausa hasta que se pulsa una tecla

World Series Baseball



Imagine/Erbe
48K
Tipo de juego: Deportivo.
P.V.P.: 1.800

El beisbol no es un juego ni mucho menos popular en España, sin embargo, estamos seguros de que este programa va a contribuir bastante a conseguir que esto deje de ser así.

El juego en cuestión es una creación de la reaparecida Imagine, la empresa de software que tan popular se hiciera en los primeros tiempos del Spectrum, y que ahora vuelve con un programa de corte deportivo en el cual se han utilizado la más sofisticadas técnicas de programación.

Como su propio nombre indica se trata de un partido de beisbol, jugado entre dos equipos diferentes, uno de ellos controlado por el ordenador, que por cierto hay que decir que lo hace bastante bien, y el otro por nosotros. Todo lo que se necesita saber es cómo se juega al beisbol, y enseguida se puede pasar a la ardua tarea de intentar controlar el partido. El manejo de nuestro equipo no es nada complicado aunque por supuesto, hay que controlar la situación con la habilidad suficiente para evitar que el contrario se aleje rápiamente de nosotros en el marcador.

La pantalla se encuentra distribuida de una forma que, además de orginal, resulta a la vez bastante práctica. De un lado, tenemos una visión general del terreno de juego, los jugadores y el público; y por otro, un vídeo marcador gigante, donde al igual que ocurre en los que se encuentran en los campos reales, podemos ver la jugada de una forma ampliada. Se ve cómo el pitcher arroja la pelota y el bateador se dispone a golpearla.

Al principio nosotros controlamos al bateador, y nuestra misión es la de mover a nuestro equipo intentando sumar carreras, mientras que después nuestra misión será defensiva y consistirá en tratar de evitar eso mismo.

En el intermedio aparecen unas majorets que recorren







el campo animando a su equipo, y que al igual que ocurría durante el juego, aparecen ampliadas en el videomarcador. También podemos ver los consabidos letreros publicitarios.

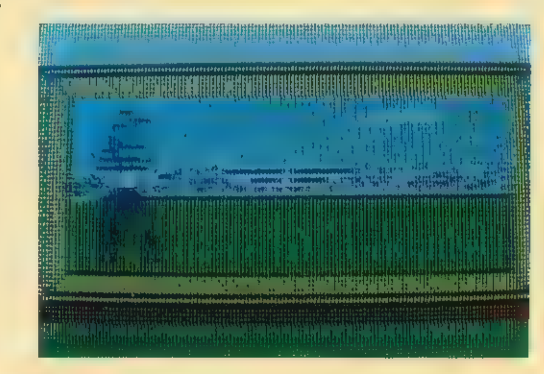
Valoración. Del juego se pueden decir tres cosas: está bien hecho, la idea es original y el nivel de adición muy alto.

Por lo que se refiere al sonido hay que destacar una simpática musiquilla que suena cuando aparecen las majorets, y el himno americano al principio.

Existe la posibilidad de jugar contra otro jugador que no sea el ordenador, lo cual es aconsejable al principio cuando no tenemos experiencia.

Muy divertido, y además un buen motivo para aprender a jugar el beisbol, sin cansarse demasiado.

Originalidad	* * * *
Gráficos	* * * *
Movimientos	* * * *
Sonido	* * *
Valoración	* * * *





LABERINTOS: CÓMO GENERARLOS

Paco MARTIN

Cuántas veces, dejando volar la imaginación no habremos pensado en crear un laberinto para el más fantástico y emocionante juego jamás construido por nosotros.

Nos imaginamos escenas llenas de emoción y misterio a raudales en los que, tras una lucha o una persecución, más o menos encarnizada, el héroe (nosotros por supuesto), llega a su destino glorioso y feliz.

Y de pronto, nos caemos de la nube, ¿cómo diablos se puede hacer tan fantástico laberinto?

Imaginamos mil y una maneras, pero él se resiste a mostrarnos sus secretos. Al final, resignados (y bastante enfadados, por qué no decirlo) decidimos hacer una especie de «laberinto aleatorio», cuyo resultado final provoca comentarios ciertamente aleatorios.

Pues bien: ¡Se acabó tal tortura! En este artículo vamos a tratar de explicar, de la forma más clara y concisa posible, su verdadero secreto. Un secreto que sólo poseían hasta ahora los mejores juegos: «FRED», «MAZIACS», etc.

Quisiéramos, antes que nada, aclarar que la forma que tendrá cada laberinto (su estructura) es algo que está en función del gusto personal de cada uno, y que el programa que vamos a explicar a continuación, es una simple muestra para entender lo único realmente importante: el «truco», «secreto», o como lo querais llamar.

Vamos a enumerar inicialmente, las condiciones más importantes y ABSOLUTAMENTE NECESARIAS para su construcción:

1. SIEMPRE DEBERA TENER DIMENSIONES IMPARES, sean éstas las que sean, para evitar que se produzcan errores durante el chequeo en la fase de construcción.

2. EL PUNTO DE PARTIDA puede decidirse aleatoriamente, pero SIEMPRE SERA IMPAR (en ambas coordenadas) y, EN NINGUN CASO DEBERA POSICIONARSE ENCIMA DE LOS LATERALES DEL LABERINTO (error en la fase de chequeo).

3. LOS PASOS DE AVANCE durante la construcción, SERAN SIEMPRE DE DOS EN DOS o de lo contrario, os podréis llevar sorpresas desagradables (normalmente el programa o se queda «atrapado» en el laberinto o hace una «masacre» que, si se está trabajando en código máquina, destruye el trabajo realizado). La mejor forma de crear un laberinto, es hacerlo directamente sobre la memoria, esto es, almacenándolo como bytes (CODE), pero para mayor claridad, lo vamos a hacer sobre una matriz de caracteres (A$ en el programa), de forma que cualquiera pueda trabajar con ella sin mayores problemas, junto con otra matriz numérica que nos servirá como «indicador de posibles caminos». Para empezar, vamos a utilizar varios convenios:

a) Los «MUROS» exteriores se han de señalizar con un byte específico para ese cometido; en nuestro caso, lo vamos a hacer con el carácter de espacio (CHR$ 32).

b) El «INTERIOR» del laberinto se llenará inicialmente con un byte indicador de «posible camino». El encargado de esta misión es el carácter UDG «A» que permitirá dar la forma de «pared» que se nos antoje.

c) El «CAMINO» que vamos a ir abriendo se irá marcando con otro byte, siendo el carácter elegido en este caso el gráfico «1» por ejemplo.



Construyamos el laberinto

Con todo esto, vamos a empezar a construir el laberinto tratando de seguir y explicar los pasos del programa.

En primer lugar se asignan sus dimensiones (línea 20) o se define su estructura global, ya que no necesariamente éste debe tener una forma definida, y a continuación, se «llena» por completo el laberinto con los bytes indicadores de «muro» (CHR$ 32, línea 38) y «posible camino» (gráfico «A», línea 40 combina ambos).

En la línea 50 se inicializan una matriz y una variable numérica (c y CO respectivamente), cuyo cometido será explicado en breve.

El «punto de partida» lo determinan las variables «V» y «H» que aquí (línea 60), toman un valor fijo pero que, como ya se dijo, éstas pueden inicializarse aleatoriamente (no olvidar las restricciones).

Para una mejor compresión del proceso, el laberinto, además de crearse en la variable A$, a su vez se va imprimiendo en la pantalla (la parte «más visible») haciéndose una impresión «general» con la instrucción GOSUB 9000.

Entramos ahora en el bucle principal en el que SE HAN DE SEGUIR los siguientes pasos:

PASO 1. (GOSUB 400). Esta sencilla pero muy importante subrutina, determina el número de «caminos libres» que podemos tomar asignando su valor a la variable LI. Si observais la figura 1, partiendo de la posición actual y siempre mirando DOS PASOS hacia la posición a examinar, se efectúa un chequeo en las cuatro direcciones posibles siendo el número obtenido igual al de «posibles caminos» o, dicho de otro modo, igual al de CHR$ (144) encontrados (UDG «A»). Si LI = 0 se salta al PASO 4 (GOSUB 300) de lo contrario, continuamos con el PASO 2.

PASO 2. Si LI > (más de una posible dirección a tomar), entonces se almacenan las posiciones vertical y horizontal (GOSUB 200) en la matriz C con subíndice CO. Así, a C (CO,1) se le asigna la posición vertical y a C (CO,2) la horizontal; luego, incrementamos el subíndice CO y continuamos con el PASO 3.

PASO 3. (Líneas 130 a 170). Si LI > = 1 se elige aleatoriamente uno de los caminos libres (si sólo hay uno, se escoge ese, naturalmente) se avanzan DOS PASOS en esa dirección llenando este espacio con dos CHR$ (143) (gráfico «1»), y se actualizan las coordenadas vertical y horizontal asignándoles la nueva posición. Nuestro programa se encarga de pintar el nuevo camino para saltar, a continuación, al PASO 1.

PASO 4. (Línea 310). El subíndice CO se reduce en uno, asignándose las nuevas posiciones vertical y horizontal con C (CO,1) y C (CO,2), respectivamente. El programa, al retornar de la subrutina, chequea si ambas son CERO. ¿Por qué?, bien, al inicializar el programa, cuando dimensionamos C (), al subíndice CO le asignamos un valor de DOS para que C (1,1) y C (1,2) NO SEAN ALTERADOS por el programa (quedando por tanto igual a CERO) ya que precisamente, estos valores se van a utilizar como INDICADORES (podían haber tomado cualquier otro, 255 por ejemplo, siempre y cuando sólo se utilicen en C () para ese cometido) y su finalidad es muy simple: cuando V= 0 y H=0 quiere decir que el laberinto... ¡está terminado! El programa salta entonces a la línea 500.

Cada vez que queramos tener una «vista» general del laberinto (el construido por este programa por supuesto), no tenemos más que hacer un GOSUB 9000 y entonces, se nos mostrará, como dijimos antes, la parte más «visible», esto es, la «parte interna del laberinto» y no sus «muros». Esto solo depende de los planes de cada uno y no hay razón alguna para no tratarlos como un elemento más (laberintos de estructura irregular, o conectados a otros minilaberintos, etc.).

Cuando queramos «movernos» por él, lo haremos chequeando el contenido de A$ (pos. horizontal). Entonces, si éste es un carácter «1», quiere decir que estamos encima de un camino y si no es así, es que hemos «tropezado» con «algo» que, o bien es un muro (UDG «A»), o bien puede ser otra «cosa»: un tesoro, un monstruo, etc. Estos últimos, claro está, se habrán introducido previamente de una forma más o menos definida, según nuestro criterio personal, siendo vuestro cometido el dar «vida» al laberinto.

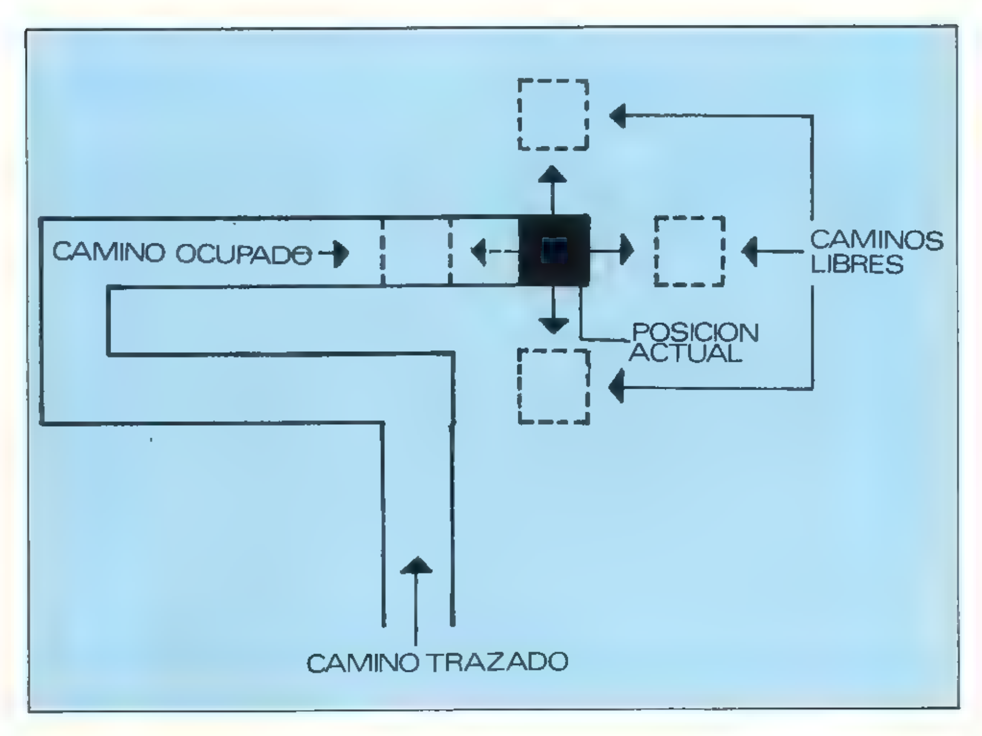
Para los que saben código máquina

Los pasos de construcción del laberinto son exactamente los mismos: se delimitará una zona en memoria inicializándose todos los bytes, cada uno con su valor específico (según sean «muros» o «posible camino»).

El STACK se utilizará para almacenar las posiciones vertical y horizontal cada vez que se chequee más de una dirección posible de avance, por lo que deberéis tener muy presente que cuanto más grande sea el laberinto, más espacio debéis reservar para el STACK si no queréis llevaros la consabida sorpresa.

Al comenzar a construir el laberinto, se inicializará el STACK, pusheando dos bytes indicadores de «LABERINTO TERMINADO» cuyo valor sólo se utilice para ese cometido (00 o ff, por ejemplo).

Cada vez que chequeemos más de un camino, pushearemos las coordenadas vertical y horizontal y las popearemos si el chequeo nos da cero, quedando terminado el laberinto cuando sus valores sean los definidos como indicadores de «fin».



¡No a los laberintos «aburridos»!

Con el método explicado, nuestro laberinto quedará totalmente «lleno» (lo cual está muy bien), pero no será muy original hacer varios laberintos. Para darle un toque «personal», bastará con rellenar (antes de que empiece a construirse el camino) el interior con «bloques» de bytes, caracteres, que impidan que se «abra» camino por allí. Insertar las líneas siguientes;

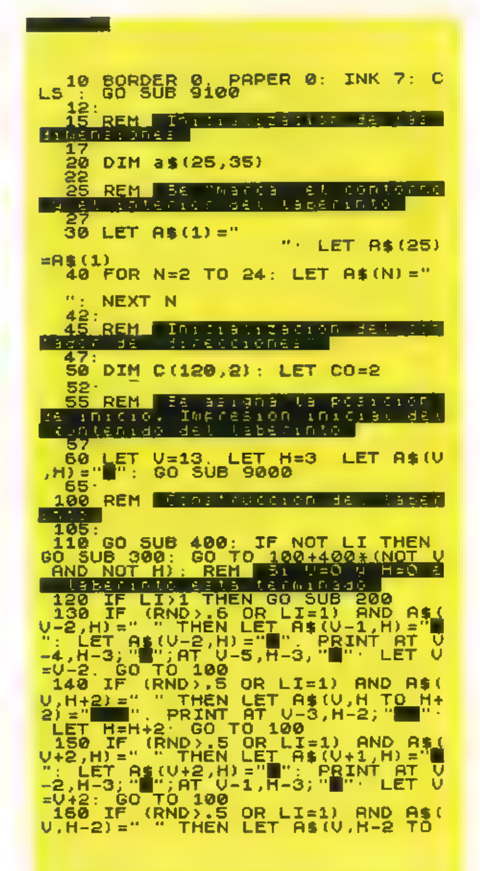
```
  52 GO SUB 9000: GO SUB 2000
2000 FOR n=1 TO 40. LET v=3+INT
(RND*19). LET h=3+INT (RND*29):
LET a$(v,h TO h+1)="BB": PRINT A
T v-3,h-3,"BB";AT v-2,h-3;"BB":
NEXT n: RETURN
```

Hacer «RUN». ¿Curioso no? La razón de utilizar los UDG «A» y «B» no es otra que permitir a los menos experimentados la posibilidad de dar unaforma cualquiera a las «paredes» y luego, poderlo visualizar de una forma sencilla.

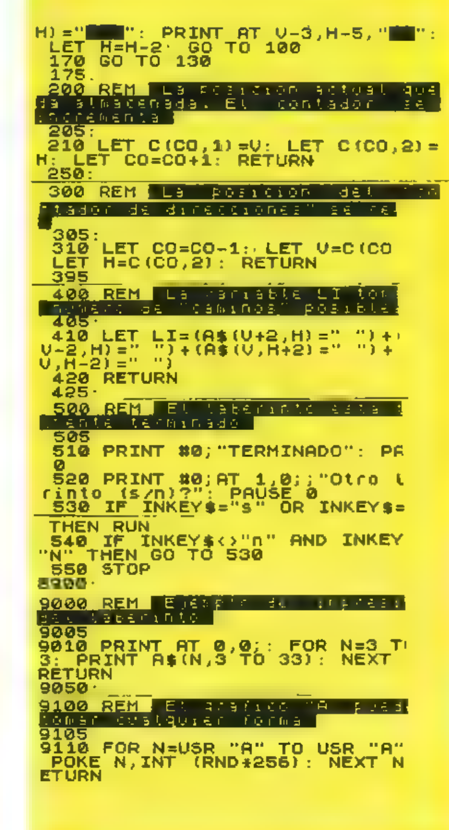
ya que la forma más correcta (siempre que se pueda) de «trabajar» con un laberinto es la de utilizar el contenido de éste como simples indicadores para hacer, a continuación, un «volcado» por bloques en pantalla del gráfico correspondiente: si detectamos un «camino» pintaremos un bloque gráfico diseñado como camino; si detectamos un muro, pintaremos el bloque correspondiente, etc. La pantalla se definirá entonces por bloques (su tamaño dependerá del grado de «aplicación» que querais darle) siendo el central la posiçión que estamos ocupando.

Esperamos que vuestros héroes se sientan orgullosos (que no acobardados) de llevar a cabo sus más gloriosas gestas entre los terribles y sinuosos muros del laberinto.

PROGRAMA GENERADOR DE LABERINTOS



```
  10 BORDER 0. PAPER 0: INK 7: C
LS : GO SUB 9100
  12:
  15 REM Inicializacion de las
dimensiones
  17
  20 DIM a$(25,35)
  22
  25 REM Se "marca" el contorno
y el interior del laberinto
  27
  30 LET A$(1)="
                  ". LET A$(25)
=A$(1)
  40 FOR N=2 TO 24: LET A$(N)="
  ": NEXT N
  42:
  45 REM "Inicializacion del con
tador de direcciones"
  47:
  50 DIM C(120,2): LET CO=2
  52·
  55 REM Se asigna la posicion
de inicio. Impresion inicial del
 contenido del laberinto
  57
  60 LET V=13. LET H=3  LET A$(V
,H)="█": GO SUB 9000
  65·
 100 REM Construccion del laber
into
 105:
 110 GO SUB 400: IF NOT LI THEN
GO SUB 300: GO TO 100+400*(NOT V
 AND NOT H): REM Si V=0 y H=0 el
  laberinto esta terminado
 120 IF LI>1 THEN GO SUB 200
 130 IF (RND>.6 OR LI=1) AND A$(
V-2,H)=" " THEN LET A$(V-1,H)="█
": LET A$(V-2,H)="█". PRINT AT V
-4,H-3;"█";AT V-5,H-3,"█"· LET V
=V-2. GO TO 100
 140 IF (RND>.5 OR LI=1) AND A$(
V,H+2)=" " THEN LET A$(V,H TO H+
2)="███". PRINT AT V-3,H-2;"███"·
 LET H=H+2· GO TO 100
 150 IF (RND>.5 OR LI=1) AND A$(
V+2,H)=" " THEN LET A$(V+1,H)="█
": LET A$(V+2,H)="█": PRINT AT V
-2,H-3;"█";AT V-1,H-3;"█"· LET V
=V+2: GO TO 100
 160 IF (RND>.5 OR LI=1) AND A$(
V,H-2)=" " THEN LET A$(V,H-2 TO
```



```
H)="███": PRINT AT V-3,H-5,"██":
 LET H=H-2· GO TO 100
 170 GO TO 130
 175.
 200 REM La posicion actual que
da almacenada. El contador se
incrementa
 205:
 210 LET C(CO,1)=V: LET C(CO,2)=
H: LET CO=CO+1: RETURN
 250:
 300 REM La posicion del "con
tador de direcciones" se re
 305:
 310 LET CO=CO-1: LET V=C(CO
 LET H=C(CO,2): RETURN
 395
 400 REM La variable LI tom
numero de "caminos" posible
 405·
 410 LET LI=(A$(V+2,H)=" ")+
V-2,H)=" ")+(A$(V,H+2)=" ")+
V,H-2)=" ")
 420 RETURN
 425·
 500 REM El laberinto esta
mente terminado
 505
 510 PRINT #0;"TERMINADO": PA
0
 520 PRINT #0;AT 1,0;;"Otro l
rinto (s/n)?": PAUSE 0
 530 IF INKEY$="s" OR INKEY$=
 THEN RUN
 540 IF INKEY$<>"n" AND INKEY
"N" THEN GO TO 530
 550 STOP
8900·
9000 REM Ejemplo de impresion
del laberinto
9005
9010 PRINT AT 0,0;: FOR N=3 T
3: PRINT A$(N,3 TO 33): NEXT
RETURN
9050·
9100 REM El grafico "A" puede
tomar cualquier forma
9105
9110 FOR N=USR "A" TO USR "A"
 POKE N,INT (RND*256): NEXT N
ETURN
```

EL CAMIONERO

José Luis SELVI

Spectrum 48 K

Somos, en esta ocasión, un sufrido camionero que se encuentra en una difícil tesitura al tener que encontrar una mercancía que desconoce y transportarla a un lugar que tiene que situar en el mapa hispano.

Así pues, nuestro ordenador nos coloca en una de las 47 capitales españolas para encaminarnos hacia una ciudad desconocida y recoger una mercancía para transportar a otra ciudad igualmente desconocida. Ante esta difícil situación, debemos utilizar un radar que lleva incorporado nuestro camión y que nos indicará la dirección en que se encuentra nuestro objetivo y la distancia que nos separa de él.

Contamos con indicadores de gasóleo y toda la información de la ciudad donde nos encontramos y a la que podemos dirigirnos por tener carretera. La red de carreteras, a la que también tenemos acceso, tiene un trazado fijo para todas las partidas, variando la otra parte para hacerlo más ameno.

Las tres ciudades, por su parte, son seleccionadas al azar y varían, lógicamente, de una partida a otra.

Para manejar el programa sólo tenemos que ir introduciendo el número de la ciudad a la que queremos dirigirnos. Pruébalo y comprobarás que, además de entretenido, harás un amplio repaso a la geografía peninsular.

NOTAS GRAFICAS

N	ñ

```
  10 GO SUB 8000  GO SUB 3000
  20 LET re=0: GO SUB 3170: GO SUB 770
  30 RANDOMIZE : GO SUB 2000: GO SUB 9000
  40 GO SUB 480  GO SUB 380
  50 IF da=0 AND cu=cm THEN GO SUB 90
  60 IF da=1 AND cu=ce THEN GO TO 210
  70 GO TO 40
 100 FOR y=0 TO 6· PRINT PAPER 1,AT y,0,c$,"          ": NEXT y
 110 PRINT INK 6,AT 0,0;"BRAVO !  Ya tienes"  PRINT "la mercancia en"  PRINT "tu poder."
 120 LET q$="          ": PRINT PAPER 1;AT 9,0;q$,AT 13,0,q$,AT 14,0;q$,AT 15,0,q$,AT 16,0;q$, PAPER 7; BRIGHT 1;AT 16,21,"          "
 130 FOR p=1 TO 40· BEEP .005.p. BEEP .007,p+2  NEXT p
 140 PRINT INK 6,AT 3,0;"Ahora la tienes": PRINT "que llevar al". PRINT "punto de destino."
 150 FOR p=40 TO 1 STEP -1. BEEP .005,p  BEEP .007.p+2  NEXT p
 160 FOR p=1 TO 40  BEEP .005,p  BEEP .007,p+2  NEXT p
 180 LET da=1
 190 RETURN
 220 PAPER 6: BORDER 6: INK 1: CLS
 230 PRINT AT 4,0;"Fenomenal !!! Lo has conseguido."
 250 FOR p=1 TO 20: BEEP .02,20  BEEP .01,25  NEXT p
 270 PRINT : PRINT "Has llevado la mercancia desde ": FOR w=6 TO 18: IF w>10 AND a$(cm,w)=" " THEN GO TO 290
 280 PRINT a$(cm,w);: NEXT w
 290 PRINT " hasta ";: PRINT a$(ce)(6 TO )
 310 PRINT : PRINT "Has necesitado ",pu;" viajes."
 320 PRINT : PRINT "Has recorrido ";INT (re*18.4)," km."
 330 LET gas=(6000-re*18.4)/100  IF gas>=0 THEN PRINT : PRINT "Han sobrado ";INT gas;" l. de gasoleo."
 340 PRINT   PRINT : PRINT "Juegas otra vez ?  (s o n)"
 350 IF INKEY$="n" THEN STOP
 360 IF INKEY$="s" THEN RUN 15
 370 GO TO 350
 380 REM Introduce ciudad a la que se va
 390 PRINT INK 1; PAPER 7;AT 21,0,"  ▪ que ciudad quieres ir ?  ";
 400 INPUT "Numero de ciudad ? ",j: IF j>47 OR j<1 THEN BEEP .7,0. GO TO 400
 410 PRINT PAPER 1,AT 21,0;"                              ": LET ban=0· LET xac=z(j,5): LET yac=z(j,6)
 420 FOR k=1 TO 4: IF z(cu,k)=j THEN LET ban=1
 430 NEXT k
 440 IF ban=0 THEN PRINT AT 21,0;"No hay carretera a ";a$(j)(6 TO ): BEEP 1.5,25· GO TO 380
 450 LET cu=j
 460 GO SUB 7000
 470 RETURN
 490 REM Calculo de direccion y distancia
 500 IF da=0 THEN PRINT INK 6,AT 0,0,"Tienes que localizar"  PRINT "la mercancia que"  PRINT "esta en alguna de": PRINT "las capitales de la": PRINT "peninsula."
 510 PRINT INK 7; PAPER 1,AT 9,0,a$(cu)(5 TO 18)
 520 PRINT AT 12,0· FOR q=1 TO 4  GO SUB 1500. IF ci=1 AND z(cu,q)>0 THEN PRINT TAB 0;a$(z(cu,q))
 530 NEXT q
 540 PRINT PAPER 1;"          "
 550 PRINT PAPER 1;"          "
 560 IF da=1 THEN GO TO 660
 570 LET fl=0. PRINT : LET mp=((ABS (z(cu,5)-z(cm,5)))↑2+(ABS (z(cu,6)-z(cm,6)))↑2)↑.5· LET mp=INT (18.4*mp): IF mp<=150 THEN LET fl=1
 600 IF da=0 THEN PRINT PAPER 7, INK 2, BRIGHT 1, FLASH fl,AT 16,21," ",mp," ", FLASH 0," Km. "
 610 LET es=500/mp
 620 LET x=es*(z(cm,5)-z(cu,5))  LET y=es*(z(cm,6)-z(cu,6))
 630 FOR o=5 TO 11: PRINT PAPER 6,AT o,23;"         ": NEXT o
 640 PLOT 212,108: DRAW INK 1;x,y
 650 RETURN
 660 LET fl=0. LET sp=((ABS (z(cu,5)-z(ce,5)))↑2+(ABS (z(cu,6)-z(ce,6)))↑2)↑.5  LET sp=INT (18.4*sp)  IF sp<=150 THEN LET fl=1
 670 IF da=1 THEN PRINT AT 16,21, PAPER 7, INK 3, BRIGHT 1; FLASH fl," ",sp," "; FLASH 0;" Km. "
 710 LET es=500/sp
 720 LET x=es*(z(ce,5)-z(cu,5)): LET y=es*(z(ce,6)-z(cu,6))
 730 FOR o=5 TO 11: PRINT PAPER 6;AT o,23;"         ": NEXT o
 740 PLOT 212,108: DRAW INK 1;x,y
 750 RETURN
 770 REM Datos de las capitales
 780 GO SUB 6000
 800 DIM a$(47,18)  DIM z(47 6)
 810 RESTORE 900  FOR h=1 TO 47  READ b$  READ b  READ c. READ d. READ e  READ f  READ g
 840 LET a$(h)=b$  LET z(h,1)=b  LET z(h,2)=c  LET z(h,3)=d  LET z(h,4)=e  LET z(h,5)=f  LET z(h,6)=g
 860 NEXT h
 870 LET da=0
 890 CLS : RETURN
 900 DATA " 1 - Albacete',b(1),28,14,a(1),42,20," 2 - Alicante",43,a(2),14,b(2),51,16," 3 - Almeria",16,a(3),28,b(3),35,2," 4 - Avila",34,8,b(4),a(4),26,34
 920 DATA " 5 - Badajoz",12,19,a(5),b(5),12,19," 6 - Barcelona",15,23,a(6),b(6),66,41," 7 - Bilbao",b(7),35,a(7),36,35,54," 8 - Burgos",4,44,a(8),b(8),32,47
 940 DATA " 9 - Caceres",a(9),b(9),34,0,17,25,"10 - Cadiz",38,27,a(10),b(10),17,1,"11 - Castellon",43,40,a(11),b(11),53,27 "12 - Ciudad Real",20,42,a(12),b(12),30,20
 960 DATA "13 - Cordoba",20,5,a(13),b(13),25,13,"14 - Cuenca",1,a(14),2,b(14),41,29,"15 - Gerona",19,6,a(15),b(15),69,45 "16 - Granada",3,27,a(16),b(16),33,6
 980 DATA "17 - Guadalajara",b(17),41,a(17),0,35,34,"18 - Huelva",a(18),5,b(18),38,12,8,"19 - Huesca",47,15,a(19),b(19),51,44,"20 - Jaen",b(20),13,12,a(20),31,11
1000 DATA "21 - La Coruña",33,25,b(21),a(21),5,55,"22 - Leon",31,46,a(22),b(22),22,49,"23 - Lerida",6,40,a(23),b(23),56,40,"24 - Logroño".45,39,b(24),a(24),39,47
1020 DATA "25 - Lugo",21,46,a(25),b(25),11,53,"26 - Madrid",b(26),37,a(26),42,32,32,"27 - Malaga",16,a(27),10,b(27),27,3,"28 - Murcia",1,3,b(28),a(28),47,12,"29 - Orense",b(29),33,a(29),30,8,47
1030 DATA "30 - Oviedo',29,36,a(30),b(30),20,55,"31 - Palencia ,a(31),44,22,b(31),27,45,"32 - Pamplona",35,45,a(32),b(32),44,50,"33 - Pontevedra",21,29,a(33),b(33),4,49
1040 DATA "34 - Salamanca",4,9,a(34),b(34),20,35,"35 - San Sebastian",7,32,a(35),b(35),41,54,"36 - Santander",30,7,a(36),b(36),30,55,"37 - Segovia",b(37) 39,26.a(37),29,35
1050 DATA "38 - Sevilla",10,b(38),18,a(38),18,8,"39 - Soria",24,37,b(39),a(39),39,42,"40 - Tarragona",23,11,b(40),a(40),62,39."41 - Teruel",47,b(41),17,a(41),47,31,"42 - Toledo",26.12,a(42),b(42),29,28
1060 DATA "43 - Valencia",a(43),2,b(43),11,51,23,"44 - Valladolid",31,8,b(44),a(44),26,42,"45 - Vitoria",b(45).24,a(45),32,38,50,"46 - Zamora",25,a(46),22,b(46),20,39,"47 - Zaragoza",a(47),19,41,b(47),48,41
1500 LET ci=1: IF z(cu,q)=0 THEN LET ci=0. RETURN
1510 IF z(cu,1)<>0 THEN IF q=2 AND a$(z(cu,q))=a$(z(cu,1)) THEN LET ci=0: RETURN
1520 IF q=3 AND z(cu,1)<>0 AND z(cu,2)<>0 THEN IF (a$(z(cu,q))=a$(z(cu,1)) OR a$(z(cu,q))=a$(z(cu,2))) THEN LET ci=0: RETURN
1530 IF q=4 AND z(cu,3)<>0 AND z(cu,1)<>0 AND z(cu,2)<>0 THEN IF (a$(z(cu,q))=a$(z(cu,3)) OR a$(z(cu,q))=a$(z(cu,1)) OR a$(z(cu,q))=a$(z(cu,2))) THEN LET ci=0: RETURN
1600 RETURN
2000 REM Situa mercancia y destino
2050 LET pu=0  LET cm=INT (46*RND+1)
2060 LET ce=INT (46*RND+1)
2070 IF ce=cm THEN GO TO 2060
2080 LET cu=INT (46*RND+1)
2090 IF cu=cm THEN GO TO 2080
2100 LET xan=z(cu,5): LET yan=z(cu,6): RETURN
3000 REM Instrucciones
3050 PRINT AT 0,11;"CAMIONERO"
3100 PRINT AT 2,1;"Te encuentras en una ciudad de la España peninsular y tienes  que recoger una mercancia en unaciudad que desconoces y llevarlaa otra, tambien desconocida."
3110 PRINT AT 8,1;"En el cuadro de instrumentos detu camion aparecen todos los da-tos que necesitas y solo tienes que introducir el numero de la  ciudad a la que quieres ir."
3120 PRINT AT 14,1;"El mapa de carreteras es un tanto original y no se parece en nada al del MOPU, por lo que te recomiendo que te vayas haciendo  el tuyo propio."
3150 LET n=0: PRINT AT 21,0,"PULSA UNA TECLA PARA EMPEZAR."
3160 LET n=n+1: IF INKEY$="" THEN GO TO 3160  RETURN
3170 CLS · PRINT AT 11,5; FLASH 1;" UN MOMENTO, POR FAVOR "
3190 RETURN
6000 REM Trazado de carreteras
6010 DIM b(47)· DIM a(47)· FOR s=1 TO 47
6020 LET b(s)=0· LET a(s)=0: NEXT s
6030 FOR s=1 TO 47
6040 LET x=INT (46*RND+1)
6050 IF x<=s THEN GO TO 6080
6060 IF RND>.55 THEN GO TO 6080
6070 IF a(x)<>0 OR a(s)<>0 THEN GO TO 6080
6080 LET a(s)=x  LET a(x)=s
6090 NEXT s
6120 FOR s=1 TO 47
6130 LET x=INT (46*RND+1)
6140 IF x<=s THEN GO TO 6180
6150 IF RND>.75 THEN GO TO 6180
6160 IF b(x)<>0 OR b(s)<>0 THEN GO TO 6180
6170 LET b(s)=x  LET b(x)=s
6180 NEXT s
6190 RETURN
7000 REM Calculo de distancia y gasoleo
7020 LET dis=((ABS (xac-xan))↑2+(ABS (yac-yan))↑2)↑.5
7030 LET re=re+dis
7040 LET xan=xac: LET yan=yac
7050 LET pu=pu+1
7060 FOR w=255 TO (255-re*.268) STEP -1
7070 PLOT INK 1;w,16  DRAW INK 1,0,7
7080 NEXT w
7090 IF re*18.4>=6000 THEN GO TO 9500
7100 RETURN
8000 REM Rotulo y define ñ
8010 PAPER 6  BORDER 6  CLS
8020 PRINT AT 8,12. INK 1,"CAMIONERO',AT 12,9, INK 2,"JOSE LUIS SELVI"
8030 RESTORE 8055
8050 FOR w=0 TO 7: READ g: POKE USR "n"+w,g· NEXT w
8060 DATA 0,60,0,44,50,34,34,0
8070 OVER 1  FOR w=1 TO 40  BEEP .05,w  BEEP .008,w+3  NEXT w
8080 FOR y=0 TO 175  INK 7-y/25  BORDER 6  PLOT 0,y  BORDER 2  DRAW y,-y  BORDER 6  PLOT 0,175-y  DRAW y y  BORDER 2  PLOT 255,y  DRAW -y,-y  BORDER 6  PLOT 255,175-y  DRAW -y,y  BORDER 2  NEXT y  BORDER 7  PAPER 7  INK 1
8090 OVER 0  CLS
8100 RETURN
9000 REM Dibujo del cuadro de control del camion
9050 PAPER 1  BORDER 1: INK 6: CLS  LET c$="          "
9070 FOR y=0 TO 44
9080 LET m=(1936-y↑2)↑.5
9090 PLOT 211-m,108+y. DRAW 2*m,0
9100 PLOT 211-m,108-y  DRAW 2*m,0
9110 NEXT y
9120 PRINT PAPER 6; INK 1,AT 3,26;"N";AT 13,26;"S";AT 8,21.'O",AT 8,31."E"
9130 PRINT PAPER 7; INK 2,AT 1,21;" R A D A R "
9140 PRINT PAPER 4; INK 0,AT 15,21,"DISTANCIA :"
9150 PRINT PAPER 5; INK 01,AT 18,21," GASOLEO : "
9160 PRINT PAPER 5; INK 1,AT 7,0;" ESTAS EN : "
9170 PRINT PAPER 7; INK 3,AT 11,0,"TIENES CARRETERAS A:"
9180 PRINT PAPER 7; BRIGHT 1,AT 16,21;c$
9190 PRINT PAPER 7; BRIGHT 1;AT 19,21;c$
9200 RETURN
9500 PAPER 6  BORDER 6  INK 1: CLS
9510 PRINT FLASH 1;AT 5,0;" Agotaste el gasoleo. "
9520 PRINT : PRINT " La mercancia estaba". PRINT " en ";a$(cm)(6 TO )
9530 PRINT : PRINT " El destino era ",a$(ce)(6 TO )
9540 PRINT : GO TO 320
```



A PERERA

Premiado con 15.000 ptas.

BUSCAPAR

Francisco MOLINA

Spectrum 48 K

Premiado con 15.000 Ptas.

Si te gustan los juegos de observación y retentiva, seguro que este programa que te ofrecemos a continuación te va a encantar.

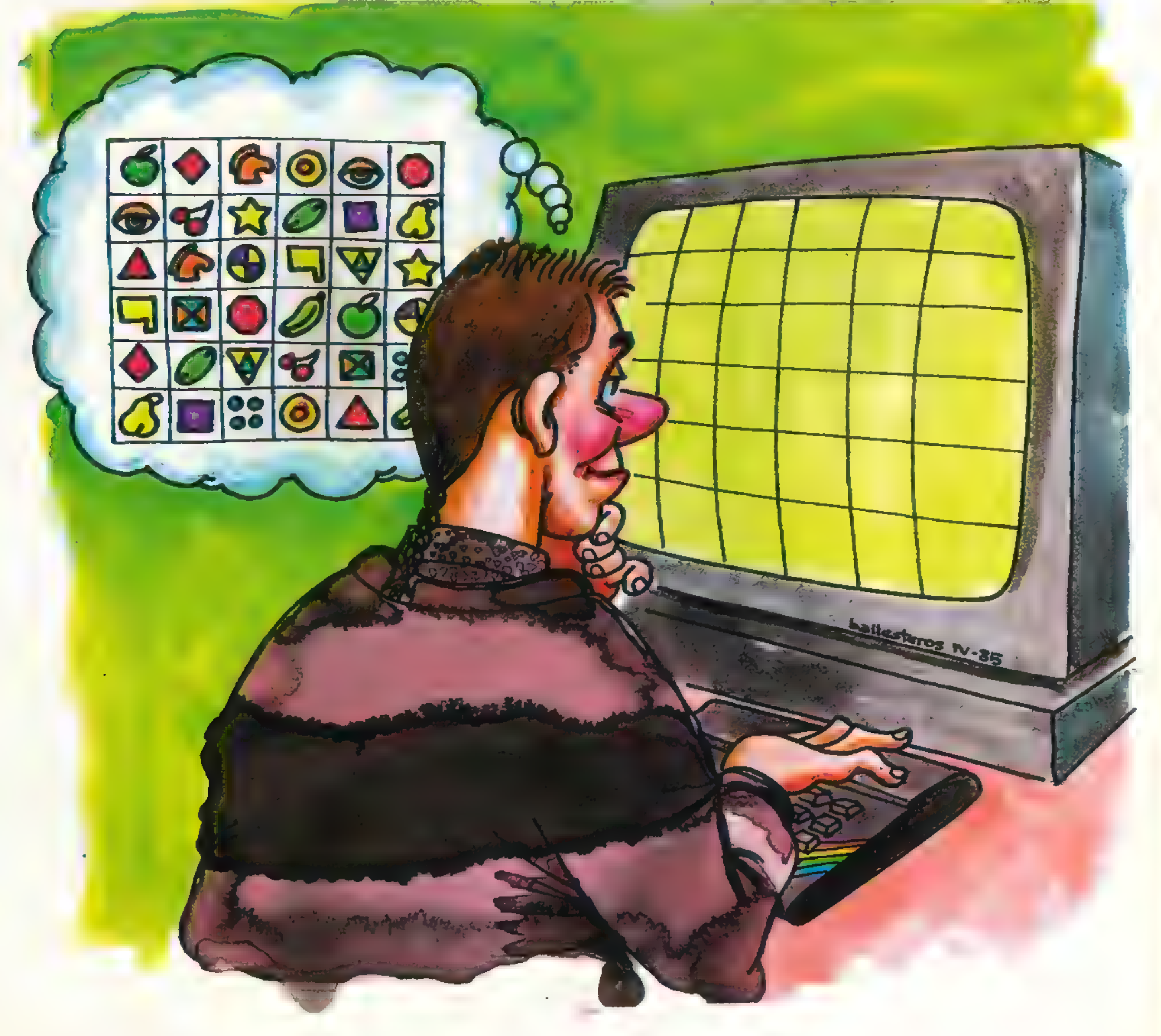
El juego consiste en memorizar los lugares que ocupan, dentro de unos casilleros, quince parejas de dibujos, y, más tarde, acertarlos. El programa empieza por colocarlos, aletoriamente, en la tabla, de tal manera que dé tiempo para su memorización. Transcurridos sesenta segundos, se borrará la pantalla y aparecerán las treinta casillas vacías con coordenadas XY, en las que se colocarán los quince dibujos y sus dobles.

Si al introducir las coordenadas de los dibujos y sus dobles, no acertamos, el ordenador ignorará los resultados y nos pedirá nuevas coordenadas. Si acertamos, aparecerán los dibujos en su casilla y unas notas musicales confirmarán nuestro tino. Inténtalo.

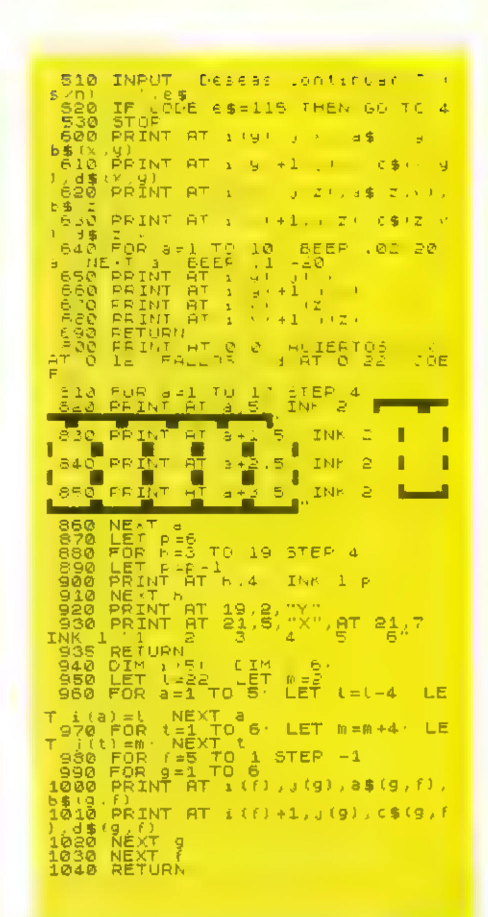
NOTAS GRAFICAS

A	B	C	D	E	F	G	H
◢	◣	◥	◤	┘	└	┐	┌

```
   1 REM ***BUSCANDO  PAREJAS***
   2 REM ***   © F. Molina    ***
   3 REM ************************
   4 CLEAR : BORDER 6: PAPER 6:
INK 0: CLS : RANDOMIZE
   5 DIM a$(6,5): DIM b$(6,5): D
IM c$(6,5): DIM d$(6,5)
   8 PRINT AT 6,6;"MEMORIZA LA P
OSICION";AT 8,9;"DE CADA DIBUJO"
;AT 10,16;"Y";AT 12,12;"SU DOBLE
"
   9 LET b=0: LET c=0: LET d=0
  10 FOR a=1 TO 15
  20 LET x=1+INT (6*RND)
  30 LET y=1+INT (5*RND)
```




```
  40 IF a$(x,y)<>" " THEN GO TO
20
  50 IF a=1 THEN LET a$(x,y)="◣"
: LET b$(x,y)="◢": LET c$(x,y)="
◤": LET d$(x,y)="◥"
  60 IF a=2 THEN LET a$(x,y)="◣"
: LET b$(x,y)="◤": LET c$(x,y)="
◢": LET d$(x,y)="◥"
  70 IF a=3 THEN LET a$(x,y)="◥"
: LET b$(x,y)="◤": LET c$(x,y)="
┘": LET d$(x,y)="└"
  80 IF a=4 THEN LET a$(x,y)="┘"
: LET b$(x,y)="└": LET c$(x,y)="
┐": LET d$(x,y)="┌"
  90 IF a=5 THEN LET a$(x,y)="┘"
: LET b$(x,y)="└": LET c$(x,y)="
◥": LET d$(x,y)="◤"
 100 IF a=6 THEN LET a$(x,y)="┘"
: LET b$(x,y)="◣": LET c$(x,y)="
┐": LET d$(x,y)="◤"
 110 IF a=7 THEN LET a$(x,y)="┌"
: LET b$(x,y)="┐": LET c$(x,y)="
└": LET d$(x,y)="┘"
 120 IF a=8 THEN LET a$(x,y)="└"
: LET b$(x,y)="┘": LET c$(x,y)="
┐": LET d$(x,y)="┌"
 130 IF a=9 THEN LET a$(x,y)="└"
: LET b$(x,y)="┘": LET c$(x,y)="
┌": LET d$(x,y)="┐"
 140 IF a=10 THEN LET a$(x,y)="◢
": LET b$(x,y)="◣": LET c$(x,y)=
"◥": LET d$(x,y)="◤"
 150 IF a=11 THEN LET a$(x,y)="◢
": LET b$(x,y)="◣": LET c$(x,y)=
"┐": LET d$(x,y)="┌"
 160 IF a=12 THEN LET a$(x,y)="◢
": LET b$(x,y)="└": LET c$(x,y)=
"◥": LET d$(x,y)="┌"
 170 IF a=13 THEN LET a$(x,y)="◢
": LET b$(x,y)="◣": LET c$(x,y)=
"┘": LET d$(x,y)="└"
 180 IF a=14 THEN LET a$(x,y)="◢
": LET b$(x,y)="◣": LET c$(x,y)=
"┌": LET d$(x,y)="┐"
 190 IF a=15 THEN LET a$(x,y)="◢
": LET b$(x,y)="◣": LET c$(x,y)=
"◢": LET d$(x,y)="◣"
 200 NEXT a
 210 LET b=b+1
 220 IF b=1 THEN GO TO 10
 230 FOR n=144 TO 151
 240 FOR f=0 TO 7
 250 READ w
 260 POKE USR CHR$ (n)+f,w
 270 NEXT f
 280 NEXT n
 290 DATA 1,3,7,15,31,63,127,255
 300 DATA 128,192,224,240,248,25
2,254,255
 310 DATA 255,127,63,31,15,7,3,1
 320 DATA 255,254,252,248,240,22
4,192,128
 330 DATA 3,3,3,3,3,3,255,255
 340 DATA 192,192,192,192,192,19
2,255,255
 350 DATA 255,255,3,3,3,3,3,3
 360 DATA 255,255,192,192,192,19
2,192,192
 361 RESTORE
 365 CLS
 370 GO SUB 810
 375 IF b=2 THEN GO SUB 940
 376 PAUSE 3000: CLS
 378 GO SUB 800
 380 INPUT "XY = ",k$," igual a
XY = ",s$
 381 IF LEN k$<>2 OR LEN s$<>2 O
R k$=s$ THEN GO TO 380
 382 IF k$(1)>CHR$ 54 OR k$(2)>C
HR$ 53 OR k$(1)<CHR$ 49 OR k$(2)
<CHR$ 49 THEN GO TO 380
 384 IF s$(1)>CHR$ 54 OR s$(2)>C
HR$ 53 OR s$(1)<CHR$ 49 OR s$(2)
<CHR$ 49 THEN GO TO 380
 386 LET x=VAL k$(1): LET y=VAL
k$(2)
 388 LET z=VAL s$(1): LET v=VAL
s$(2)
 390 LET k$="  ": LET s$=""
 400 IF a$(x,y)=" " OR a$(z,v)="
" THEN GO TO 380
 410 IF a$(x,y)<>a$(z,v) OR b$(x
,y)<>b$(z,v) OR c$(x,y)<>c$(z,v)
 OR d$(x,y)<>d$(z,v) THEN LET d=
d+1: FOR a=1 TO 10: BEEP .02,20-
a: NEXT a: BEEP .1,-20: GO SUB 6
00: PRINT AT 0,19;d: GO TO 380
 420 LET c=c+1
 430 BEEP .5,10: BEEP .25,5: BEE
P .5,20
 435 PRINT AT 0,9;c
 440 PRINT AT i(y),j(x);a$(x,y);
b$(x,y)
 450 PRINT AT i(y)+1,j(x);c$(x,y
);d$(x,y)
 460 PRINT AT i(v),j(z);a$(z,v);
b$(z,v)
 470 PRINT AT i(v)+1,j(z);c$(z,v
);d$(z,v)
 480 LET a$(x,y)=" ": LET a$(z,v
)=" "
 490 FOR a=1 TO 6: FOR q=1 TO 5:
 IF a$(a,q)<>" " THEN GO TO 380
 491 NEXT q
 492 NEXT a
 500 PRINT AT 0,27; FLASH 1;(INT
 (d/c*100))/100
```



```
 510 INPUT "Deseas continuar ? (
s/n) ";e$
 520 IF CODE e$=115 THEN GO TO 4
 530 STOP
 600 PRINT AT i(y),j(x);a$(x,y);
b$(x,y)
 610 PRINT AT i(y)+1,j(x);c$(x,y
);d$(x,y)
 620 PRINT AT i(v),j(z);a$(z,v);
b$(z,v)
 630 PRINT AT i(v)+1,j(z);c$(z,v
);d$(z,v)
 640 FOR a=1 TO 10: BEEP .02,20-
a: NEXT a: BEEP .1,-20
 650 PRINT AT i(y),j(x); [illegible]
 660 PRINT AT i(y)+1,j [illegible]
 670 PRINT AT i [illegible] (z [illegible]
 680 PRINT AT i [illegible] +1 [illegible] (z [illegible]
 690 RETURN
 700 PRINT AT 0,0;"ACIERTOS";AT 0,12;"FALLOS";AT 0,22;"COE
F"
 810 FOR a=1 TO 17 STEP 4
 820 PRINT AT a,5; INK 2;"[graphics]"
 830 PRINT AT a+1,5; INK 2;"[graphics]"
 840 PRINT AT a+2,5; INK 2;"[graphics]"
 850 PRINT AT a+3,5; INK 2;"[graphics]"
 860 NEXT a
 870 LET p=6
 880 FOR h=3 TO 19 STEP 4
 890 LET p=p-1
 900 PRINT AT h,4; INK 1;p
 910 NEXT h
 920 PRINT AT 19,2;"Y"
 930 PRINT AT 21,5;"X";AT 21,7;
INK 1;"1   2   3   4   5   6"
 935 RETURN
 940 DIM i(5): DIM j(6)
 950 LET l=22: LET m=2
 960 FOR a=1 TO 5: LET l=l-4: LE
T i(a)=l: NEXT a
 970 FOR t=1 TO 6: LET m=m+4: LE
T j(t)=m: NEXT t
 980 FOR f=5 TO 1 STEP -1
 990 FOR g=1 TO 6
1000 PRINT AT i(f),j(g);a$(g,f);
b$(g,f)
1010 PRINT AT i(f)+1,j(g);c$(g,f
);d$(g,f)
1020 NEXT g
1030 NEXT f
1040 RETURN
```

Representación de los números en el Spectrum (II)

SISTEMA HEXADECIMAL

José T. CROVETTO

La notación hexadecimal es, probablemente, la más adecuada para relacionarse con un ordenador. Permíteme representar números más grandes con menos dígitos y, por otra parte, al incluir letras en sus expresiones, resulta más legible para el usuario.

Al leer su manual del Spectrum habrá encontrado que, al tratar sobre los números, emplea la base diez (sistema decimal que utilizamos habitualmente), la base dos (sistema binario del que hablamos en el primer artículo de esta serie) y la base dieciséis o notación hexadecimal.

La primera pregunta que nos podemos hacer es, ¿por qué la base dieciséis y no otra? Si la información en la memoria del Spectrum se guarda en sistema binario (bits), ¿qué utilidad tiene que nosotros empleemos otra base para representar a los mismos números? La contestación a estas preguntas requiere conocer la base dieciséis.

Veamos, en primer lugar, la representación de números enteros en notación hexadecimal.

Tabla I

Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Hemos escrito los números binarios de la tabla con cuatro bits. Realmente el cero y el uno se pueden representar con un bit, el dos y el tres con dos bits, y el cuatro, cinco, seis y siete con tres bits. Sin embargo, recordarlo así, se verá posteriormente que resulta útil.

Así pues, un número expresado en base dieciséis puede contener letras (A a F) en alguna de sus cifras, algo a lo que no estamos muy acostumbrados.

Vistos esto, el tratamiento de números expresados en báse dieciséis es similar al de base dos o base diez que vimos en nuestro anterior artículo. Aplicando los mismos principios, tenemos que: Cualquier número en base dieciséis se representa por combinaciones de los dígitos o cifras hexadecimales, asignando un peso a cada uno de ellos según el lugar que ocupen.

El peso de cada uno de los dígitos hexadecimales que forman el número, es la potencia de dieciséis igual al lugar que ocupen, comenzando a contar desde cero y de derecha a izquierda. Por ejemplo:

En el número 734H, los pesos de los dígitos son, comenzando por el de la derecha, 16^0 (= 1), 16^1 (= 16) y 16^2 (= 256), respectivamente, es decir que: 734H = $7 \times 16^2 + 3 \times 16^1 + 4 \times 16^0 = 7 \times 256 + 3 \times 16 + 4$ = 1844D

Obsérvese como conociendo los pesos, la conversión hexadecimal a decimal es inmediata.

Veamos otro ejemplo:

¿Cuál será el equivalente decimal del hexadecimal FE2AH ?

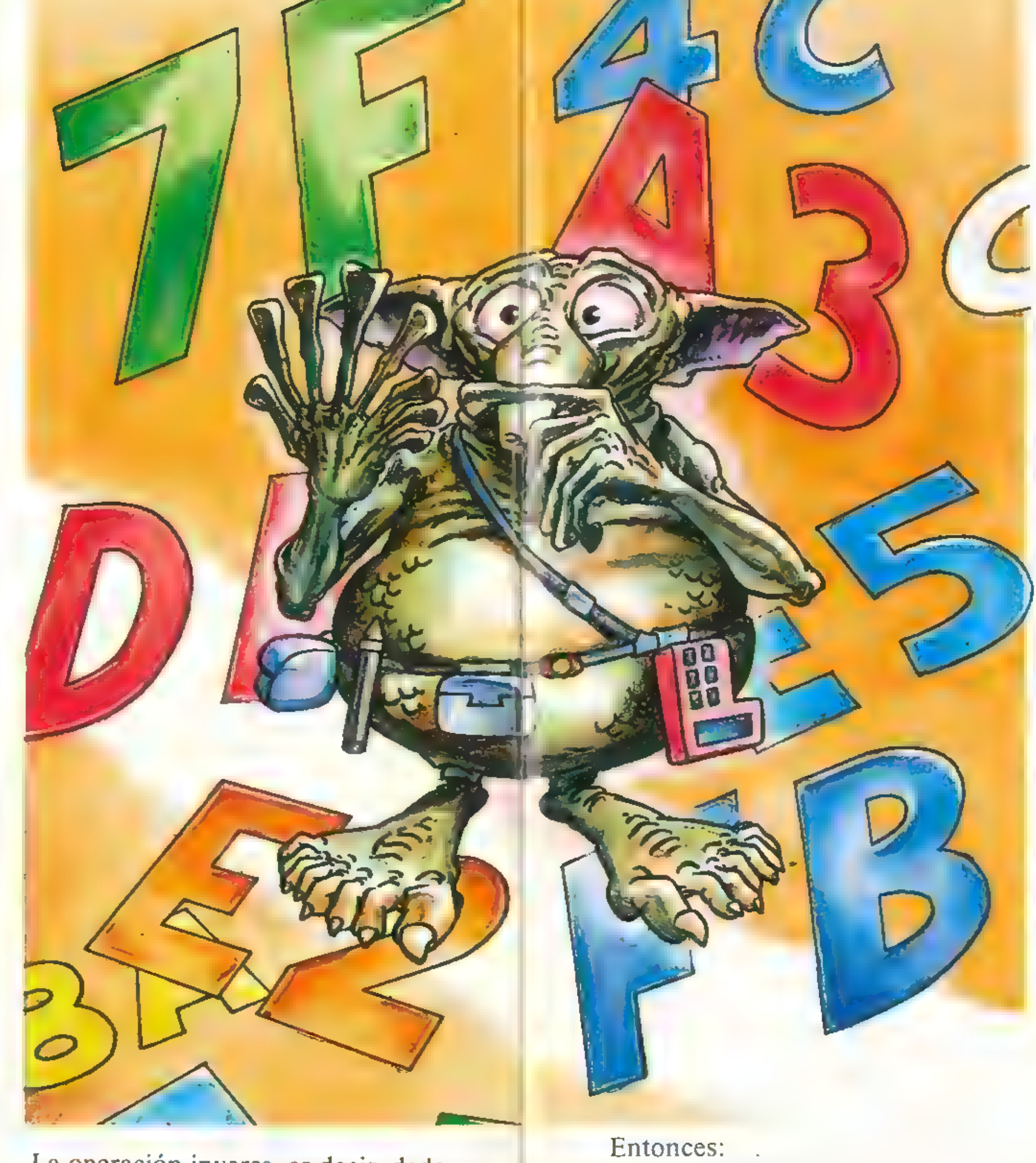
Si procedemos como en el ejemplo anterior, tendremos:

FE2AH = $F \times 16^3 + E \times 16^2 + 2 \times 16^1 + A \times 16^0 = 15 \times 4096 + 14 \times 256 + 2 \times 16 + 10$ = 65066D en base diez.

Recuerde que el equivalente decimal del dígito hexadecimal F es 15D, el de E es 14D y el de A es 10D.

Es útil aprender de memoria las cuatro primeras potencias de dieciséis, que se corresponden con los pesos de los cuatro primeros dígitos (comenzando por la derecha) de cualquier número hexadecimal:

$16^0 = 1$; $16^1 = 16$; $16^2 = 256$; $16^3 = 4096$



La operación inversa, es decir, dado un número en base diez calcular su equivalente en base dieciséis, se resuelve mediante un algoritmo, similar al visto en el anterior artículo para la conversión de base diez a base dos, denominado de divisiones sucesivas. El método consiste en dividir el número entre dieciséis y el cociente de esta división dividirlo, nuevamente, entre dieciséis, y así sucesivamente hasta que el cociente resultante sea inferior a dieciséis. Veamos un ejemplo:

Calcular el equivalente hexadecimal del decimal 61345D.

```
61345 |16
 133   3834 |16
  054   063   239 |16
   065   154   079  14
     1    10    15
```

Hemos puesto en negrita los restos de las distintas divisiones y el último cociente. El equivalente hexadecimal se construye tomando estos números empezando por el último cociente y escribiéndolos de izquierda a derecha.

Entonces:

61345D = EFA1H

Recuerde nuevamente que el equivalente hexadecimal de 14D es EH, el de 15D es FH y el de 10D es AH.

Igual que en base diez, añadir ceros a la izquierda de un número entero hexadecimal no altera el valor. El mismo número representan 34AH y 0034AH.

El siguiente programa convierte un número decimal a hexadecimal.

```
  10 REM PROGRAMA DE CONVERSION
DE DECIMAL A HEXADECIMAL
  20 LET B$=""
  30 INPUT "numero?=";A
  40 IF A=0 THEN PRINT B$· GO TO
 20
  50 LET C=INT (A/16): LET R=A-1
6*C
  60 LET C$=STR$ (R)
  70 IF R>10 THEN LET C$=CHR$ (5
5+R)
  80 LET B$=C$+B$
  90 LET A=C
 100 GO TO 40
```

Para la conversión de hexadecimal a decimal pruebe el siguiente programa:

```
  10 REM PROGRAMA DE CONVERSION
DE CUALQUIER BASE ENTRE 2 Y 16 A
  BASE 10
  20 INPUT "base?=",B: IF B<2 OR
 B>16 THEN GO TO 140
  30 INPUT "numero?=";A$  LET C=
0
  40 FOR I=0 TO LEN (A$)-1
  50 LET A=CODE (A$(LEN (A$)-I))
  60 IF (A<48 OR A>57) AND (A<65
 OR A>70) THEN GO TO 140
  70 IF A>=48 AND A<=57 THEN LET
 A=A-48: GO TO 90
  80 LET A=A-55
  90 LET C=A*(B↑I)+C
 100 IF A>=B THEN GO TO 140
 110 NEXT I
 120 PRINT C
 130 GO TO 20
 140 PRINT "ERROR": GO TO 20
```

En los dos programas anteriores, el número tecleado debe ser positivo

Hasta aquí, hemos visto la conversión binaria↔decimal y hexadecimal↔decimal. Veamos ahora la conversión directa binario↔hexadecimal.

La utilidad de la base dieciséis está precisamente en que la conversión binario — hexadecimal es sencilla, directa y rápida. Además, en base dieciséis se pueden representar con pocas cifras números que en base dos necesitarían una larga tira de «unos» y «ceros». Resulta muy cómodo referirse al valor de un byte (8 bits) o de una palabra (2 bytes = 16 bit) expresándolo en base dieciséis mejor que en binario o en decimal. Con algo de practica, es muy fácil —en seguida se llega a hacer mentalmente— conocer el valor de cada bit dentro del byte (o palabra).

Para la conversión directa de binario a hexadecimal separamos el número binario en grupos de cuatro bits, comenzando por la derecha. Si alguno faltara en el último grupo (el de la izquierda) se añaden ceros, lo que no altera el número. A continuación, convertimos cada grupo de cuatro bits a su dígito hexadecimal correspondiente (ver tabla I), y los escribimos en el mismo orden. Por ejemplo, el número 10110011111B es en hexadecimal:

0101 1001 1111B = 59FH
5 9 F

Y el número 101110100011B es en hexadecimal:

1011 1010 0011B = BA3H
B A 3

Obsévese que un byte se puede representar con dos dígitos hexadecimales puesto que:

00000000B = 00H
11111111B = FFH = 255D

La conversión de hexadecimal a binario directa consiste en hacer justo lo contrario. Es decir, cada cifra hexadecimal se convierte a su equivalente binario (Tabla I), escribiendo cada uno de los dígitos hexadecimales con cuatro bits y en el mismo orden. Por ejemplo, el hexadecimal, A31FH en binario sería:

A31FH = 1010 0011 0001 1111B
A 3 1 F

y el

3E9H = 0011 1110 1001B = 1111101001B
3 E 9

El hexadecimal:

FFFFH = 1111111111111111B = 65535D

se corresponde con la dirección más alta de la memoria del Spectrum de 64 Kbyte (1 Kbyte = 2^{10} = 1024 bytes).

CONSULTORIO

La rutina de rótulos

¿Por qué mi ordenador no carga bien la rutina de rótulos de la cinta «Horizontes»? He comprobado el listado del relocalizador y no me he equivocado en nada; sin embargo, cuando llega el momento de cargar la cinta, no funciona, como por falta de volumen.

J. F. MARTIN - Málaga

☐ Tenga en cuenta que en la cinta «Horizontes» antes del bloque de C/M que sirve para crear los rótulos, hay dos pantallas y un programa en Basic. Cuando el relocalizador está cargando la cinta ignora estos bloques, y no carga hasta que no encuentra el que tiene como nombre de fichero: «C», lo que no ocurre hasta pasados dos o tres minutos de la cara A.

Borrados parciales

¿Existe alguna sentencia o grupo de ellas que permita borrar un solo objeto de la pantalla sin que ésta se borre totalmente?

¿Cómo se logra el efecto de paso de un objeto sobre otro distintos?

¿Cómo se hacen pantallas como las del juego MUGSY?

Daniel RODRIGUEZ - MADRID

☐ Para borrar un carácter determinado de la pantalla, la mejor forma es imprimir un espacio encima de él.

El efecto de paso de un objeto sobre otro se logra imprimiendo en OVER 1.

Las pantallas como las del juego MUGSY se hacen con programas especiales para dibujar. Hay muchos en el comercio y no tendrá problemas en encontrar el más adecuado a sus necesidades.

Conservación del Spectrum

Querría saber si el zumbido de alarma que realiza cuando la línea está a tope de capacidad, es peligroso para el buen estado del ordenador.

¿Se va deteriorando el Spectrum poco a poco si se utiliza sólo para juegos?

¿Sería conveniente abrir de vez en cuando el ordenador para limpiarlo de polvo?

El teclado profesional, ¿tiene un sistema estudiado? o por el contrario estropea las teclas que el ordenador ya posee.

¿Por qué hay programas que consiguen dibujar en las dos líneas inferiores? ¿Es simplemente porque el código máquina lo permite y el Basic no?

Miguel A. ARTACHO - Logroño

☐ El zumbido de alarma no perjudica en absoluto al ordenador.

Lo único que puede deteriorarse del ordenador por su uso repetitivo es el teclado.

El ordenador sólo debe ser abierto por causas justificadas y, en todo caso, las menores veces posibles.

La mayor parte de los teclados profesionales se acoplan quitando el teclado que lleva el ordenador.

Para escribir en las líneas inferiores de la pantalla desde el Basic, utilice: PRINT# 1; (texto).

Spectrum musical

Quiero grabar un programa de música para el ZX Spectrum 48K, y antes de hacerlo quisiera saber si al Spectrum se le puede acoplar un altavoz para que se escuche más fuerte la música.

¿Se puede grabar la música en una cinta por la salida de auricular del ordenador (EAR)? Y por último, ¿se puede de acoplar unos cascos al Spectrum?

Rafael LOZANO - Barcelona

☐ Si desea acoplar un altavoz al ordenador deberá intercalar un amplificador de audio. La salida idónea es MIC.

La música producida por el ordenador se puede grabar de la misma forma que los programas.

Para acomplar unos cascos, hágalo del mismo modo que en el caso del altavoz.

Código Máquina

¿Es necesaria una programación en código máquina para cada tipo de ordenador?

Si fuera necesario, quisiera me informarais si hay algún libro específico de código máquina para el Spectrum.

Entre lenguaje máquina y lenguaje esamblador, ¿cuál de ellos tiene más ventajas?

Antonio GUANEZ - Madrid

☐ Cada microprocesador utiliza código máquina distinto, en el caso del Spectrum deberá utilizar el código máquina del microprocesador Z-80.

En el comercio encontrará amplia bibliografía sobre el código máquina del Z-80 e incluso, sobre programación en código máquina para el Spectrum.

Assembler y código máquina es un mismo lenguaje. En código máquina es el resultado de esamblar un programa escrito en Assembler.

Teclados profesionales

No entiendo qué son y para qué sirven los teclados profesionales para el Spectrum que tanto se anuncian en MICROHOBBY.

Fernando PEREZ - Madrid

☐ Se trata, como su nombre indica, de unos teclados que sustituyen al incómodo teclado de goma del Spectrum, dándole además, una mejor apariencia.

Interfaces

¿Para qué sirve el Interface 2?, ¿cuánto vale?

¿Hay algún interface para que el sonido del Spectrum se oiga?, ¿cuánto vale?

Juan C. RAMOS - Madrid

☐ El Interface 2 sirve para conectar al Spectrum dos joysticks y un cartucho de ROM.

Indescomp comercializa un amplificador de sonido para el Spectrum.

En lo relativo a precios, y dada la variabilidad de los mismos, es preferible que consulte a un distribuidor.

¡¡PROGRAMATE EL AÑO CON NOSOTROS!!

AHORA PUEDES BENEFICIARTE CON CUALQUIERA DE NUESTRAS NUEVAS FORMAS DE SUSCRIPCION

1 ☐ **SI**, deseo suscribirme a la Revista **Microhobby Semanal** durante un año (50 números), lo que me da derecho a recibir, automáticamente, como regalo, un lote de cinco cintas vírgenes especiales para ordenador, marca «Sound-on-Sound».

50 revistas por sólo **4.250 pts.**

(Ahorro 500 ptas. más un regalo de 1.100 pts.)

2 ☐ **SI**, deseo que mensualmente me sean enviadas todas las **Cintas de Programas de** Microhobby Semanal, que se editen. Esta suscripción me da derecho a un precio reducido por cada cinta, y a no abonar gastos de envío.

12 cintas por sólo **5.500 pts.**

3 ☐ **SI**, deseo recibir a su precio normal la(s) cinta(s) de Programas que indico a continuación. Cada cinta lleva grabados los programas publicados por **Microhobby**durante cuatro números consecutivos (1 al 4, 5 al 8, 9 al 12, etc.) y su precio es de **550** pts. más **75** pts. por gastos de envío cada una*.

Números del .... al ...... (inclusive)
Números del .... al ...... (inclusive)
Números del .... al ...... (inclusive)

550 pts. + 75 pts.

*En el caso de las cintas sueltas no se admiten pedidos contra reembolso ni T. de C. Por favor, envíe **talón o giro postal.**

4 ☐ **SI**, deseo beneficiarme de las ventajas de la **Suscripción Conjunta,** que supone **50 números de Microhobby Semanal y 12 Cintas de Programas,** a precio aún más reducido. Esta suscripción también me da derecho a recibir el regalo de cinco cintas para ordenador marca «Sound-on-Sound».

50 revistas y 12 cintas por sólo **8.900 pts.**

(Ahorro 3.350 ptas. más un regalo de 1.100 pts.)

NOMBRE .... EDAD ....
APELLIDOS ....
DOMICILIO ....
CIUDAD .... PROVINCIA ....
C. POSTAL .... TELEFONO .... PROFESION ....

Marco con una (X) en el casillero correspondiente la forma de pago que mas me conviene.

☐ Talón bancario adjunto a nombre de HOBBY PRESS, S.A. ☐ Giro Postal N.º .... ☐ Contra reembolso del primer numero.

TARJETA DE CREDITO: ☐ VISA VISA N.º .... ☐ MASTER CHARGE N.º ....

Fecha de caducidad de la tarjeta .... Firma:

(cortar por la linea de trazos)

Franqueo
Postal

HOBBY PRESS, S. A.

Apartado de Correos

n.º **54.062** (Apartados Altos)

MADRID

DE OCASION

● DESEARIA intercambiar programas, comentarios e ideas con chicos-as aficionados al Spectrum. Interesados llamar Tlf. (972) 573673. Preguntar por Pedro. Mn. Ramón Avellana, s/n. Mata, Bañolas (Gerona).

● VENDO vídeo juego Atari cón dos pares de mandos, transformador y 6 juegos; Phoenix, Combat, Galaxian, Pac-man, Asteroides, Defender. Todos con instrucciones. Todo por 20.000 ptas. Para pedir información llamar de 20,30 a 22,00 al Tlf. (922) 275929. Tenerife. Hora canaria, preguntar por Dionisio.

● VENDO videojuegos ATARI con dos pares de mandos y cuàtro cartuchos, entre ellos COMECOCOS, y ASTEROIDES; también incluyo instrucciones, del manejo. Todo por 10.000 ptas., y sólo tiene un año de uso. Dirigirse al Tlf. 2301511. Alejandro Avelli Montes. Barcelona.

● VENDO Home Computer Philips G-7400. A estrenar, ganado en concurso. Regalo cartucho. Precio a convenir. Llamar al Tlf. (91) 6941990, Paco a partir de las 8,30.

● VENDO Spectrum 16K, 25.000 ptas; teclado profesional, 10.000; ampliación de memoria a 48K EX., 7.000; casette, 4.000; televisor B/N portatil, 7.000; impresora GP-505, 22.000; T.R.Q. estabilizador 125 V. y 220 V., 5.000. En conjunto, 75.000 ptas. con regalo de cintas y libros de más de 10.000. José Barredo Sampedro. Marcelino González, 12, P. 2.°C. Tlf. 143551, Gijón (Asturias).

● VENDO una computadora personal Casio, modelo P. B. 100, económica, o cambio por Spectrum, 48K, con garantía, abonando cantidad razonable. Interesados llamar al Tlf. 3400473. Valencia. Vicente (de 2 a 4 de la tarde).

● COMPRO ZX Spectrum 48K, con garantía. Interesados preguntar por Vicente, Tlf. 3400473. Valencia (de 2 a 4 de la tarde).

● VENDO sintetizador de voz Currah Microspeech, con dos semanas de uso y en perfecto estado, por 9.900 ptas. Incluye amplificador de sonido por el televisor, manual de instrucciones en inglés y cinta de demostración. Compatible con la mayoría de programas comerciales. Llamar a Victor, Tlf. 2767997 (Madrid).

● VENDO Spectrum 48K, monitor B/N de 12", programas, manuales completos de instrucciones, aún tiene 10 meses de garantía el Spectrum, todo por 50.000 ptas. También VENDO completísimo laboratorio fotográfico de B/N, precio a consultar. Jesús Suárez Gutiérrez. Ramón y Cajal, 45. La Robla (León). Tlf. (987) 570037.

● VENDO un sintetizador de voz Currah por 7.000 ptas, además, vendo 150 programas para Spectrum (Orc Attack, Dr. Franky, etc.) por 7.500 ptas. También me interesa el intercambio de programas, dirigirse a Pedro Morales. Mediodía, 68 P. 3° plta. 306. Lloret del Mar (Gerona). Tlf. 369246.

● CAMBIO programas para el Spectrum 16/48K. Clemente. Escultor Salas, 14, 1.°. 50007 Zaragoza. Tlf. (976) 372426.

● VENDO ordenador Vic 20 con accesorios. Escribir a Luis Miguel Ortega Gil. Gordoniz, 66, 6.° D. 48002 Bilbao, o llamar al Tlf. (94) 4440064 preguntando por Luis Miguel.

● CAMBIO ordenador ZX Spectrum 48K completamente nuevo más 10.000 ptas., por un Commodore 64. Si le interesa, escriba a Roger Mayola Castillo. Dos de Mayo, 327. 08025 Barna. Tlf. 2358294.

● VENDO ZX Spectrum de 16K, nuevo, con garantía. Más la cinta de «Horizontes», el alimentador, el manual de Basic y el de instrucciones en castellano, los cables y conexiones. Todo por sólo 25.000 ptas. M.ª del Mar Arnáez. Empecinado, 3. 47003 Valladolid. Tlf. (983) 251685.

● VENDO sintetizador de voz Currah y ZX Spectrum 48K, tres libros Spectrum y 200 programas comerciales todo con sus accesorios y manuales y cintas de demostración por 35.000 ptas. (todo incluido). Llamar al Tlf. (972) 369246, noche. También cambio programas por periféricos. Escribir a Pedro Morales. Mediodía 68, P. 3.°, plta. 306. Lloret del Mar (Gerona).

● VENDO ordenador Spectrum Plus, totalmente nuevo, aún embalado y con toda su garantía de seis meses. Se incluyen todos los cables, transformador de corriente, manual, instrucciones en castellano y estupendos juegos como: TLL, VU-3D, Manic Miner, Trans-Am, Stop the Express, Ajedrez, etc. Todo ello por 45.000 ptas. Fco. Javier de Antonio. General Aranda, 14, 1.° B. 28029 Madrid. Tlf. 215.97.40.

● VENDO Spectrum 16K con instrucciones, cables, fuente de alimentación y 4 cintas complementarias. Todo en perfecto estado. Comprado en el 84, pero con menos de 22 horas de trabajo. Por 24.000 ptas. Llamar al Tlf. (924) 500047, o escribir a Carlos Moñino. Queipo de Llano, 2. Fuente de Cantos (Badajoz).

● VENDO ZX Spectrum 16K, 6 cintas, manuales, adaptador y cables. Fecha de compra 19-7-84. 26.000 ptas. Javier Revenga Fernández. Pintor Zurbarán, 5, 3.° A. Tlf. 6179991.

CONCURSO MASTER-MIND

Ponemos en conocimiento de nuestros lectores que el pasado día 15 de abril finalizó el plazo de admisión de cintas para el concurso Master-Mind que, organizado por MICROWORLD y MICROHOBBY, había ampliado el plazo de admisión, previsto en un principio para el 28 de febrero.

A partir de este momento y en las próximas semanas, iremos dando información sobre el desarrollo del mismo.